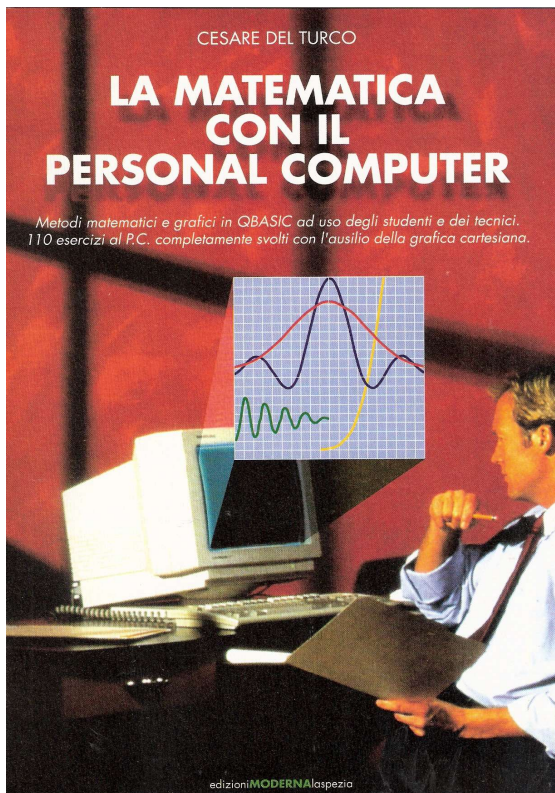
SONAR-INFO-p11

Un ausilio al calcolo numerico per le problematiche attinenti al SONAR, alla Correlazione ed altro

LA MATEMATICA CON IL PERSONAL COMPUTER

Un libro pubblicato nel 1997 per programmare in linguaggio QBASIC

Il testo in oggetto consente un facile approccio al calcolo numerico mediante personal computer. Con 110 esercizi numerici e grafici, completamente sviluppati, è utile per acquisire un minimo d'esperienza necessaria per l'elaborazione delle funzioni matematiche per le più diverse applicazioni tecniche.

Il libro è redatto in linguaggio QBASIC che, se pur superato da VISUAL BASIC, resta sempre un valido e semplice strumento per le applicazioni matematiche.

L'utilizzo del testo è possibile in due modi:

1) Per chi utilizza Visual Basic, ricordando che gli operatori matematici dei due linguaggi sono identici, non resta che convertire il vecchio linguaggio al nuovo, da file.bas a file.frm, scegliendo tra i numerosi esercizi quelli ritenuti più utili.

2) Per coloro che desiderano cimentarsi con QBASIC è possibile operare direttamente scaricando l'eseguibile QBASIC.exe dalla rete adattandolo al proprio sistema operativo.
Con l'ausilio del QBASIC.exe il testo guida l'utilizzatore dalle prime semplici operazioni agli sviluppi più complessi.

Per agevolare l'utilizzo del testo sono disponibili numerosi file, per il calcolo e la grafica, che evitano al lettore la digitazione dei molti listati stampati sul libro per l'esecuzione degli esercizi. I file, del tipo file.bas, sono nominati con i corrispondenti numeri di figura che compaiono nel testo, ad esempio Fig78.bas per l'ultimo esercizio che genera la figura 78; l'insieme dei file è contenuto nella cartella fig.zip
Non tutte le figure, purtroppo, hanno il corrispondente file, quelli disponibili sono scaricabili cliccando su: figure.

L'indice e il testo sono consultabili in file.pdf, capitolo per capitolo, cliccando:

INDICE
Cap.1 Modalità di interfaccia e strumenti matematici
Cap.2 Esercitazioni numeriche di programmazione
Cap.3 La grafica e gli esercizi per la visualizzazione delle funzioni matematiche
Cap.4 La soluzione delle equazioni algebriche trascendenti
Cap.5 La ricerca dei punti notevoli delle funzioni
Cap.6 Derivate di funzioni
Cap.7 Integrazione definita delle funzioni
Cap.8 Gli algoritmi di Fourier
Cap.9 Gli algoritmi di correlazione
Cap.10 I polinomi di butterworth e di chebychev e le loro trasformazioni
Cap.11 I numeri complessi
Appendice.1 Applicazioni del QBASIC per l'analisi dei quadripoli
Appendice.2 La soluzione di un problema di geometria analitica

Home

INDICE

CAPITOLO 1

CAPITOLO 2
ESERCITAZIONI NUMERICHE DI PROGRAMMAZIONE

CAPITOLO 3

LA GRAFICA E GLI ESERCIZI PER LA
VISUALIZZAZIONE DELLE FUNZIONI MATEMATICHE

CAPITOLO 4

LA SOLUZIONE DELLE EQUAZIONI ALGEBRICHE
E TRASCENDENTI

CAPITOLO 5

LA RICERCA DEI PUNTI NOTEVOLI DELLE FUNZIONI

CAPITOLO 6

DERIVATE DI FUNZIONI

CAPITOLO 7

INTEGRAZIONE DEFINITA DELLE FUNZIONI

CAPITOLO 8

GLI ALGORITMI DI FOURIER

CAPITOLO 9

GLI ALGORITMI DI CORRELAZIONE

CAPITOLO 10

I POLINOMI DI BUTTERWORTH
E DI CHEBYCHEV E LE LORO TRASFORMAZIONI

CAPITOLO 11

I NUMERI COMPLESSI

APPENDICE. 1

APPLICAZIONI DEL QBASIC PER LANALISI DEI QUADRIPOLI

APPENDICE. 2

LA SOLUZIONE DI UN PROBLEMA DI GEOMETRIA ANALITICA

CAPITOLO 1
MODALITA' DI INTERFACCIA E STRUMENTI MATEMATICI

Questo capitolo del testo mostra il semplice passaggio tra l'ambiente MS-DOS versione 6.22 e il Qbasic in esso installato, introduce alle modalità di interfaccia tra il linguaggio Qbasic e l'operatore per consentire l'impiego della tecnica di programmazione oggetto del nostro impegno, fornisce inoltre gli strumenti per la compilazione dei programmi matematici.

1.1 Passaggio al programma operativo Qbasic e il ritorno in MS-DOS

Quando accendiamo il P.C., dopo le verifiche automatiche svolte dal sistema operativo MS-DOS, sullo schermo compare la scritta:
C :\> che indica che il P.C. è pronto a ricevere le istruzioni dell'operatore alla tastiera, a questo punto per entrare in ambiente Qbasic basta digitare la scritta "qbasic" a fianco della precedente come segue:
C:\> qbasic e premere il tasto INVIO (ENTER)
avremo la comparsa sul video della schermata di benvenuto che faremo scomparire premendo il tasto ESC, resterà sullo schermo la finestra blu dentro la quale andremo ad operare in linguaggio Qbasic.
La finestra operativa è corredata da un certo numero di scritte per la gestione del programma che sono disposte, come mostrato in figura 1, sia nella parte alta dello schermo, sia nella parte bassa. Per il nostro lavoro impiegheremo prevalentemente la scritta "File" che compare in alto a sinistra.

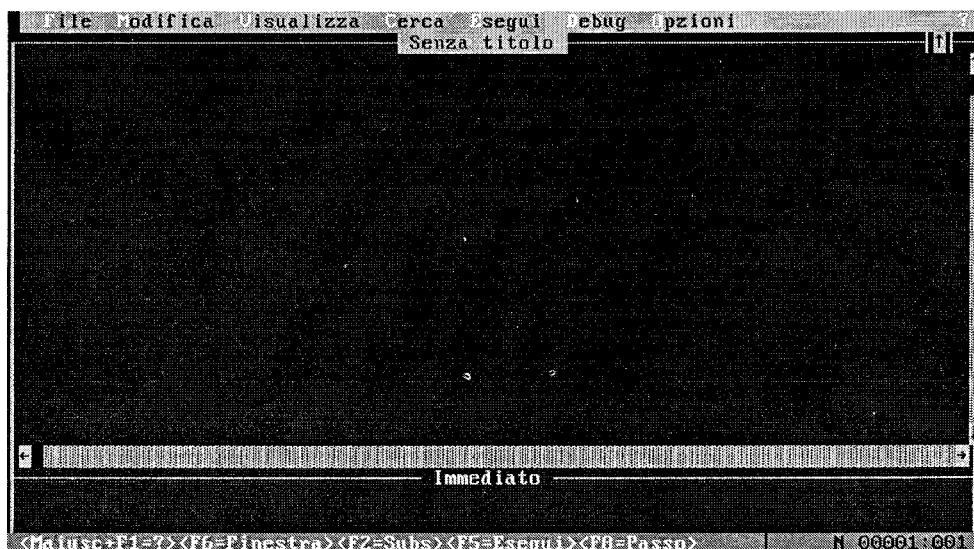



Figura 1
Finestra operativa

Se clicchiamo con il mouse sulla scritta "File" compare nella parte alta dello schermo una lista di operazioni di gestione che possiamo eseguire prima o dopo aver compilato i nostri programmi in Qbasic; queste sono:

Nuovo
Apri....
Salva
Salva con nome ...
Stampa.....
Esci

Il modo di impiego delle diverse operazioni di gestione, ad eccezione della prima e l'ultima, sarà spiegato nel prosieguo del testo quando se ne renderà necessario l'utilizzo.
Se clicchiamo con il mouse sulla scritta **Nuovo** diamo al P.C. l'informazione che il lavoro che ci accingiamo a svolgere è nuovo ed è al momento senza titolo, la conseguenza di questa azione fa scomparire la lista di gestione dallo schermo e lo stesso è pronto ad accogliere il nostro lavoro.
Compare in questo caso, in alto al centro dello schermo, la scritta "Senza titolo".

Per ritornare in MS-DOS basta cliccare con il mouse su **Esci.**

1.2 Le variabili matematiche

Con l'ausilio di poche righe di testo rinfreschiamo alcune nozioni fondamentali presenti senz'altro nel bagaglio culturale di ciascun lettore.
In tutti gli sviluppi matematici, siano essi indirizzati alla soluzione di problemi tecnici, siano essi destinati a studi inerenti l'analisi matematica o la geometria analitica, sono definite **VARIABILI** quelle grandezze, convenzionalmente denotate con le lettere X e Y, che sono scuscettibili di assumere valori diversi per descrivere numericamente ogni tipo di processo.
Le variabili che hanno caratteristiche di indipendenza si dicono **VARIABILI INDIPENDENTI.**
Le variabili che hanno caratteristiche di dipendenza da altre variabili si dicono **VARIABILI DIPENDENTI.**
Se scriviamo la **notazione esplicita simbolica** di **FUNZIONE** $Y = f(x)$ indichiamo che **x** è la variabile indipendente mentre **Y**, che dipende da **x**, è la variabile dipendente (la notazione $Y = f(x)$ indica infatti che Y è funzione di x).
Lo stesso concetto di **funzione** si riprende mediante una delle innumerevoli **notazioni esplicite strutturate** quale il calcolo del seno di un arco:

$$Y = \text{Sen}\ x$$

dove **x**, variabile indipendente, rappresenta l'argomento (arco od angolo) della funzione trigonometrica, mentre **Y** rappresenta come varia la funzione seno (variabile dipendente) al variare dell'argomento **x** .
La notazione esplicita simbolica di una funzione $Y = f(x)$ indica genericamente una qualsiasi funzione di x, mentre la notazione esplicita strutturata si riferisce ad una ben determinata funzione descritta mediante una espressione matematica.
Ricordiamo inoltre che le funzioni possono essere legate a più variabili indipendenti; in questo caso la loro notazione esplicita simbolica assumerà la forma:

$$Y = f(X_1 \ ; \ X_2 \ ; ..... ; \ X_n)$$

Alcune funzioni dette "parametriche" sono legate tanto alla variabile indipendente quanto a parametri caratteristici che generano le "famiglie di funzioni", la notazione simbolica di una funzione parametrica è del tipo:

$$Y = f(X1 \; ; \; X2 \; ; \; A)$$

dove con X1 e X2 sono indicate ad esempio due variabili indipendenti e con la lettera A è indicato il parametro. Le funzioni fondamentali che costituiscono le basi della trigonometria e dell'analisi matematica sono dette "funzioni elementari".
Le funzioni necessarie per la soluzione dei più diversi problemi possono essere semplici funzioni elementari, algoritmi formati da più funzioni elementari o da funzioni di funzioni.

1.3 Estensione del concetto di funzione

Per facilitare l'uso del linguaggio matematico di Qbasic è utile estendere i concetti che abbiamo ricordato al paragrafo precedente anche a operazioni di aritmetica elementare che normalmente non sono proposte come funzioni.
Se ad esempio consideriamo l'operazione di somma tra due addendi X1; X2 il risultato si scrive:

$$S = X1 + X2$$

dove gli addendi si possono configurare come variabili indipendenti ed il risultato S della loro somma si può configurare come la variabile dipendente.
La somma indicata si può pertanto scrivere con la notazione esplicita simbolica:
$S = f(X1 \; ; X2)$
intendendo in questo caso che la funzione S è dipendente dalle due variabili X1 e X2 .
Questo semplice modo di ragionare è naturalmente estensibile, sia alle altre tre operazioni elementari quali la sottrazione, la moltiplicazione e la divisione, sia alle operazioni di potenza e di radice quadrata; complessivamente sei funzioni che possono essere rappresentate normalmente mediante la lettera Y come variabile dipendente secondo le seguenti notazioni esplicite strutturate:

somma	$Y = X1 + X2$
sottrazione	$Y = X1 - X2$
moltiplicazione	$Y = X1 \cdot X2$
divisione	$Y = X1 : X2$
potenza	$Y = X^n$
radice q.	$Y = \sqrt{X} = (X)^{1/2}$

1.4 Le istruzioni per il colloquio con Qbasic

Dopo aver avuto accesso alla schermata di lavoro del linguaggio Qbasic, seguendo le indicazioni del paragrafo 1.1, è ora necessario imparare le semplici istruzioni con le quali poter **comandare**, quando di bisogno, **l'ingresso e l'uscita dati** dai nostri programmi per colloquiare con Qbasic.

1.4.1 Le istruzioni di ingresso dati

Prima di digitare le istruzioni di ingresso dati è conveniente digitare sulla schermata l' istruzione:

CLS

Digitando come prima istruzione CLS si assicura la pulizia automatica dello schermo in modo che risultati di elaborazioni di calcolo precedenti non compaiano assieme a risultati correnti creando confusione di lettura.
Le **istruzioni di ingresso dati** che, quando necessario, devono precedere il programma di lavoro sono relative alla richiesta che il Qbasic dovrà fare all'operatore affinché possa introdurre, digitandoli, i valori da assegnare alle variabili indipendenti ed ai parametri; questi valori sono alla base di tutte le elaborazioni di calcolo in Qbasic perché in esso vengono sviluppate sempre funzioni di una o più variabili e a volte funzioni parametriche.
Il numero delle variabili indipendenti e dei parametri che possono essere introdotti non ha limitazioni pratiche; se ad esempio si devono impiegare due variabili e due parametri: X1, X2, A, C, si dovranno digitare nell'ordine le seguenti quattro istruzioni separate da (invio):

```
INPUT " X1 " ; X1
INPUT " X2 " ; X2
INPUT " A " ;  A
INPUT " C " ;  C
```

Ciascuna istruzione contiene la scritta di richiesta INGRESSO DATI (**INPUT**),
la scritta che definisce la variabile (**" X1 "**), ad esempio, che permette, al momento della richiesta da parte di Qbasic, di far comparire sul video la lettera stessa accompagnata da un punto interrogativo, la scritta (**; X1**) che impone al valore che sarà digitato di inserirsi nel programma di calcolo previsto.
E' utile a questo punto fare un semplice esempio pratico con il P.C. su quanto abbiamo detto: supponiamo pertanto di voler iniziare un programma di lavoro, che non necessariamente dobbiamo ora digitare, che richieda l'introduzione del valore di una variabile indipendente X1 e del parametro B, inizieremo a digitare in sequenza e premere il tasto invio come segue:

```
CLS                 (invio)
INPUT " X1" ; X1    (invio)
INPUT " B" ; B      (invio)
```

dopo l'ultimo (invio) premeremo il tasto **F5** per il passaggio obbligato all'esecuzione delle tre istruzioni che abbiamo digitato. A seguito della pressione su F5 scompariranno le istruzioni digitate in campo blu e comparirà, in campo nero, la richiesta di introduzione del valore della variabile indipendente con la scritta:

X1 ?

collocata nella parte alta a sinistra dello schermo.
Supponiamo ad esempio che il valore da assegnare ad X1 sia 9.5, digiteremo tale valore a fianco di X1? come segue:

X1 ? 9.5 (invio)

a seguito del comando invio si avrà la comparsa della scritta in B:

B ?

Con questa azione il valore 9.5 è stato introdotto in un ipotetico programma in Qbasic, e la seconda scritta ci chiede il valore che vogliamo assegnare al parametro B; se questo deve essere per esempio -3.75 digiteremo tale valore ottenendo sullo schermo:

B ? - 3.75 (invio)

premendo (invio) il valore -3.75 verrà introdotto nell'ipotetico programma in Qbasic, contemporaneamente comparirà in fondo allo schermo la scritta (**premere un tasto per continuare**), si preme nuovamente **F5** e si ritorna alla schermata blu di partenza in cui sono ancora presenti le istruzioni digitate all'inizio. La sequenza può essere ripetuta quante volte sia necessario per l'introduzione di altre serie di valori che la variabile indipendente ed il parametro debbano assumere.
Con questo esercizio abbiamo preso confidenza con le modalità di introduzione dei valori delle variabili indipendenti e dei parametri che possono essere richiesti da un qualsiasi tipo di programma di lavoro.

1.4.2 Le istruzioni di uscita dati

Le istruzioni di **uscita dati** da un programma collocate, quando necessario, alla fine dello stesso hanno lo scopo di rendere all'operatore i risultati dei calcoli eseguiti dal programma di lavoro.
I risultati dei calcoli sono naturalmente i valori assunti dalle variabili dipendenti in funzione dei valori assegnati alle variabili indipendenti ed ai parametri.
Il numero delle funzioni elaborate nell'ambito di un programma di calcolo non ha limitazioni pratiche; se ad esempio sono state implementate nel programma di lavoro sette diverse funzioni i risultati di sette diverse elaborazioni numeriche devono essere presentate all'operatore, ciascuna distinta dalle altre mediante lettere o combinazioni di lettere e numeri diversi.

1.4.2.1 Le istruzioni di uscita dati per presentazione in video

Le notazioni delle funzioni possono essere ad esempio tre Y1, Y2, Y3, e le istruzioni necessarie, da collocare alla fine del programma di lavoro, per rendere possibile la lettura sul **video** dei risultati dovranno essere digitate come segue:

```
PRINT " Y1 = " ; Y 1
PRINT " Y2 = " ; Y2
PRINT " Y3 = " ; Y3
```

Ciascuna istruzione contiene la scrittura di richiesta di stampa del valore della variabile dipendente (**PRINT**), contiene inoltre la scrittura che definisce la variabile che dovrà essere presentata sul video con il segno di uguale, ad esempio (**" Y1 = "**), infine la scrittura (**; Y1**) che permette al programma di andare a prelevare i valori da presentare nell'ambito del complesso di calcolo di Y1.
Così come fatto per le istruzioni di INPUT costruiamo un esempio di applicazione pratica per le istruzioni di PRINT.
Supponiamo che un ipotetico programma abbia già calcolato i valori delle tre variabili dipendenti e questi siano Y1 = 3.6 Y2 = -1.45 Y3 = 0.45, per inserire nel nostro esempio questi tre valori, per poter impiegare le istruzioni finali di PRINT, è necessario digitare, prima delle tre istruzioni di PRINT, i valori stessi come se fossero tre righe di istruzioni così come è sotto riportato, avendo cura di premere il tasto (invio) dopo aver digitato ciascun valore di Y e dopo ciascuna delle tre istruzioni di PRINT.

Y1 = 3.6 (invio)
Y2 = -1.45 (invio)
Y3 = .45 (invio) (il programma segue nella pag. 12)

PRINT " Y1 = " ; Y1 (invio)
PRINT " Y2 = " ; Y2 (invio)
PRINT " Y3 = " ; Y3 (invio)

dopo aver digitato quanto sopra premeremo il tasto **F5** per passare all'esecuzione delle tre istruzioni di PRINT. A seguito della pressione su F5 scompariranno le istruzioni ora scritte su campo blu e compariranno, su campo nero, i risultati attesi nella seguente forma:

Y1 = 3.6
Y2 = -1.45
Y3 = .45

comparirà inoltre sulla parte bassa del video la scritta (**premere un tasto per continuare**).
Premendo F5, od un altro tasto a piacere, si tornerà alla schermata blu di programma.
E' opportuno osservare che nell'esercizio ora eseguito le prime tre righe digitate portano di fatto le identiche scritte che compaiono dopo aver premuto F5, ciò può trarre in inganno dando la sensazione che le tre righe in questione vengano trasferite per la lettura per il solo fatto di averle in precedenza digitate. Ciò non corrisponde a realtà, infatti se nell'esercizio omettiamo le altre tre righe relative alle istruzioni di PRINT e premiamo F5 si ha la scomparsa delle righe rimaste ma nessun dato viene scritto in campo nero sul video, dove compare soltanto la scritta (**premere un tasto per continuare**).

1.4.2.2 Le istruzioni di uscita dati per la presentazione su stampante

Le istruzioni di fine programma che preparano la presentazione dei dati all'operatore mediante scrittura degli stessi su stampante, sono simili alle precedenti descritte per la presentazione video; la differenza tra le due consiste, per la stampa, nella anteposizione della lettera **L** all'istruzione **PRINT**.
Avremo di conseguenza che il complesso delle sei istruzioni dell'esercizio di paragrafo 1.4.2.1, per essere provato con stampante, deve essere scritto come segue:

Y1 = 3.6 (invio)
Y2 = -1.45 (invio)
Y3 = .45 (invio)
LPRINT " Y1 = " ; Y 1 (invio)
LPRINT " Y2 = " ; Y2 (invio)
LPRINT " Y3 = " ; Y3 (invio)

dopo aver digitato quanto sopra premeremo il tasto **F5** per passare all'esecuzione delle tre istruzioni di LPRINT. A seguito della pressione su F5 scompariranno le istruzioni ora scritte su campo blu e saranno scritti sulla stampante i risultati attesi nella forma:

Y1 = 3.6
Y2 = -1.45
Y3 = .45

Premendo F5 si tornerà alla schermata blu di programma .

1.4.2.3 La presentazione contemporanea dei dati su video e stampante

La presentazione dei valori delle variabili dipendenti è possibile contemporaneamente sia su video che su stampante, per far ciò si devono digitare sia le istruzioni PRINT che le istruzioni LPRINT così come riportato nell'esercizio seguente:

```
Y1 = 3.6                   (invio)
Y2 = -1.45                 (invio)
Y3 = .45                   (invio)
PRINT " Y1 = " ; Y 1       (invio)
PRINT " Y2 = " ; Y2        (invio)
PRINT " Y3 = " ; Y3        (invio)
LPRINT " Y1 = " ; Y 1      (invio)
LPRINT " Y2 = " ; Y2       (invio)
LPRINT " Y3 = " ; Y3       (invio)
```

dopo aver digitato quanto sopra premeremo il tasto **F5** per passare all'esecuzione delle istruzioni di PRINT e LPRINT. A seguito della pressione su F5 scompariranno le istruzioni scritte su campo blu e saranno scritti sul video e sulla stampante i risultati attesi nella forma:

```
Y1 = 3.6
Y2 = -1.45
Y3 = .45
```

Premendo F5 si tornerà alla schermata blu di programma .

1.5 L'uso del tasto invio

Nei paragrafi precedenti abbiamo visto che il tasto invio deve essere premuto dopo aver digitato ciascuna istruzione, per evidenziare ciò abbiamo avuto cura di scrivere accanto ad ogni istruzione tale dizione raccolta tra parentesi (invio)**; dato che questa operazione è sempre necessaria, nel prosieguo del testo non indicheremo più, per semplicità, questa informazione che sarà ricordata dal lettore.**

1.6 Esercitazione completa di programmazione

In questo paragrafo riportiamo una semplicissima ma completa esercitazione di programmazione in linguaggio Qbasic che può essere provata dal lettore sul P.C. Si fissa come obiettivo la elaborazione numerica della funzione somma indicata nel paragrafo 1.3:

$$Y = X1 + X2$$

questa funzione ha il vantaggio, su altre, che può essere scritta in linguaggio Qbasic così come si scrive comunemente in modo algebrico.

Si stabilisce che la funzione debba essere computata per le seguenti serie di coppie della variabile indipendente:

X1	X2
3	7
- 6.7	1.5
1.95	- 4.3

(si osservi che la scrittura dei numeri decimali deve essere espressa con il punto di separazione e non con la virgola; con la stessa espressione saranno presentati i valori della variabile indipendente Y)
La stesura del programma inizia con le istruzioni di cui al paragrafo 1.4.1, a queste segue la funzione somma ed in chiusura l'istruzione di cui al paragrafo 1.4.2.1 così come sotto indicato:

```
CLS
INPUT " X1 " ; X1
INPUT " X2 " ; X2
Y = X1 + X2
PRINT " Y = " ; Y
```

dopo l'ultimo (invio) premiamo il tasto **F5** per il passaggio obbligato all'esecuzione delle istruzioni che abbiamo digitato. A seguito della pressione su F5 scompaiono le istruzioni digitate in campo blu e compare, in campo nero, la richiesta di introduzione del valore della prima delle variabili indipendenti con la scritta:

X1 ?

che andiamo a completare digitando il primo valore di X1 = 3:

X1 ? 3

a seguito del comando invio si ha la comparsa della nuova richiesta:

X2 ?

Con questa azione il valore 3 è stato introdotto nel programma in Qbasic e la seconda scritta ci chiede il valore che vogliamo assegnare a X2 che nella nostra tabella è stato fissato in X2 = 7. Digitiamo tale valore e otteniamo sullo schermo:

X2 ? 7

premiamo (invio) ed il valore 7 viene introdotto nel programma in Qbasic, contemporaneamente compare il primo valore calcolato della variabile dipendente Y nella forma:

Y = 10

ed in fondo allo schermo la scritta **(premere un tasto per continuare)**, si preme nuovamente **F5** e si ritorna alla schermata blu di partenza in cui sono ancora presenti le istruzioni digitate all'inizio.
La sequenza si ripete per la coppia X1 = - 6,5; X2 = 1.5 con il risultato:
Y = -5
ed ancora per la coppia X1 = 1.9; X2 = - 4.3 con il risultato:
Y = -2.35 .
La banalità dell'operazione e dei risultati non devono condurre il lettore ad una sottovalutazione del lavoro svolto; è opinione dell'autore che un approccio molto semplice alle problematiche della programmazione di carattere matematico serva a preparare la via ai più complessi processi di calcolo che nel prosieguo del testo dovranno essere affrontati.

1.7 Commenti ai programmi

E' molto utile, a volte indispensabile, affiancare alle istruzioni o a gruppi di istruzioni un commento scritto che riporta il significato delle istruzioni stesse.
Se in un programma molto semplice il commento può risultare superfluo, nel caso di programmi complicati rappresenta l'unico mezzo con il quale, a distanza di tempo, si può ricostruire la filosofia di programmazione seguita che è caratteristica peculiare e distintiva di ciascun programma.
Il commento al programma è cosa estremamente semplice; si tratta di affiancare all'istruzione digitata il significato dell'istruzione stessa separandolo dalla prima mediante il segno ' disponibile sul tasto in cui è presente anche il simbolo (?) (tastiera italiana).
Con queste informazioni procediamo, a solo titolo dimostrativo e non per necessità, al commento del programma presentato nel paragrafo 1.6.

```
( programma )              (commento)

CLS                        ' istruzione per la pulizia dello schermo

INPUT " X1 " ; X1          ' istruzione per l'ingresso della prima variabile indipendente

INPUT " X2 " ;  X2         ' istruzione per l'ingresso della seconda variabile indipendente

Y = X1 + X2                ' funzione da elaborare

PRINT " Y = " ; Y          ' istruzione per la presentazione video del valore della
                           ' variabile dipendente
```

Come si vede per l'ultima istruzione sono state richieste due righe di commento; in questo caso ciascuna riga deve essere preceduta sempre dal segno ' ; **se il commento richiede più righe queste devono essere sempre precedute dal segno '** .
E' importante osservare che il commento dei programmi non altera minimamente i risultati di questi ma porta ad un allungamento dei tempi di esecuzione degli stessi da parte del P.C. Naturalmente questo fenomeno non è assolutamente avvertibile dall'operatore; ci sono però casi in cui i programmi dedicati ad elaborazioni di dati che pervengono al P.C. dall'esterno possono risentire delle perdite di tempo dovute ai commenti, in questi casi particolari potrà essere necessario eliminare i commenti stessi dal programma che diventa operativo.

1.8 La rivelazione degli errori di compilazione

In Qbasic è implementato un sistema per la rivelazione degli errori che l'operatore può commettere durante la digitazione delle diverse istruzioni che compongono un programma.
Se durante la compilazione del programma si commette un errore qualsiasi, dalla mancanza di una parentesi, ad un errore di sintassi, allo scambio tra una virgola ed un punto ed altro, al primo (invio) che si preme compare, sovrapposta al testo digitato, una finestra con indicazioni accompagnata da un rettangolino luminoso sopra o nei pressi della presenza dell'errore.
Per avere un'idea della situazione conseguente ad uno degli innumerevoli errori che si possono commettere supponiamo di aver digitato l'istruzione non corretta

```
INPUT " A = " . A
```

invece dell'istruzione corretta

INPUT " A = " ; A

dove l'errore consiste nell'aver digitato un punto invece che un punto e virgola prima della lettera A. In questo caso appena premeremo (invio) per passare a digitare l'istruzione seguente comparirà sullo schermo un rettangolino luminoso sul punto errato ed una finestra con le seguenti indicazioni:

Era atteso : ; o ,

< OK> < GUIDA>

Le informazioni date nella finestra sono molto chiare: dove si è acceso il rettangolino luminoso era atteso un punto e virgola o una virgola.
A questo punto l'operatore può procedere alla correzione indicata dopo aver rimosso la finestra indicatrice mediante la pressione del tasto **ESC**.
L'esempio che abbiamo mostrato, che si suggerisce di ripetere al P.C., è uno dei più semplici ed evidenti, in molti altri casi purtroppo la ricerca dell'errore, rimarcato dal sistema di rivelazione, richiede molta pratica all'uso della "Guida per ricerca errori "che è richiamata nella finestra; guida alla quale si può accedere cliccando con il mouse sulla scritta < Guida > della finestra stessa.

1.9 Gli strumenti matematici disponibili in Qbasic

Prima di impiegare le metodologie di programmazione Qbasic è necessario conoscere gli strumenti matematici che questo tipo di linguaggio è in grado di offrire.
Le funzioni matematiche in linguaggio Qbasic devono essere, a volte, espresse secondo una simbologia diversa da quella ordinaria comunemente impiegata in trigonometria od in analisi matematica, per tale ragione sono proposte di seguito, per le funzioni elementari più usate, le corrispondenze tra le due simbologie. Sono infine trattate le memorie di calcolo che sono alla base dei programmi in Qbasic.
Le funzioni di uso poco corrente non sono riportate, per casi particolari si dovrà consultare il testo classico HANDBOOK OF MATHEMATICAL FUNCTIONS di Abramowitz -Stegun.
La simbologia che è indicata in **grassetto** è la stessa che dovrà essere digitata mediante la tastiera nelle fasi di compilazione dei programmi di lavoro, la simbologia in caratteri normali è quella comunemente impiegata nell'ambito della matematica ordinaria.
Le funzioni che sono in dotazione del linguaggio Qbasic si dicono " implementate ", le funzioni che si ottengono mediante manipolazioni matematiche a programma si dicono " implementabili ".
Le serie delle funzioni elementari, raccolte nei successivi paragrafi, serviranno come ausilio e riferimento nella stesura dei vari programmi di lavoro, a questo scopo sono numerate progressivamente per poter più facilmente richiamarle in seguito.

1.9.1 Le funzioni aritmetiche

Le funzioni aritmetiche impiegate in Qbasic sono:

Simbologia ordinaria	Simbologia Qbasic
1) (radice quadrata) (NB. Nel testo per indicare la radice quadrata verrà sempre impiegato l'esponente frazionario 1/2)	
$y = \sqrt{x} = (x)^{1/2}$	**y = SQR (x)**

2) (potenza di n qualsiasi)
$y = x^n$ **y = (x) ^ n**

3) (moltiplicazione)
$y = a \cdot b$ **y = a * b**

4) (divisione)
$y = a : b = a / b$ **y = a / b**

5) (somma)
$y = a + b$ **y = a + b**

6) (differenza)
$y = a - b$ **y = a - b**

7) (valore assoluto)
$y = | x |$ **y = ABS (x)**

1.9.1.1 Gli operatori aritmetici relazionali

Le corrispondenze simboliche relative agli operatori aritmetici relazionali sono:

Simbologia ordinaria	Simbologia Qbasic
8) (maggiore di)	
$a > b$	**a > b**
(maggiore od uguale)	
$a \geq b$	**a > = b**
9) (minore di)	
$a < b$	**a < b**
(minore od uguale)	
$a \leq b$	**a < = b**
(diverso)	
a /= b	**a < > b**

1.9.2 Funzioni trigonometriche

Le funzioni trigonometriche che sono implementate od implementabili in Qbasic prevedono l'impiego di argomenti (valori angolari) espressi in radianti; nel caso in cui gli argomenti debbano essere espressi in gradi sessagesimali è necessario trasformarli in radianti con la seguente espressione:

argomento in radianti = (**0.017453293 · A°**)

dove A ° è l'argomento in gradi sessagesimali .
Se per esempio si deve calcolare il seno di 45° (sessagesimali) il corrispondente argomento in radianti è:

x = 0.017453293 · 45° = 0.78539818

La corrispondenza tra le funzioni trigonometriche espresse in simbologia ordinaria e simbologia Qbasic implementata è la seguente:

Simbologia ordinaria	Simbologia Qbasic
10) (seno di x) y = Sen x	**y = SIN (x)**
11) (coseno di x) y = Cos x	**y = COS (x)**
12) (tangente di x) y = Tang x	**y = TAN (x)**

La corrispondenza tra le funzioni trigonometriche espresse in simbologia ordinaria e implementabili in simbologia Qbasic si ottiene:

Simbologia ordinaria	Simbologia Qbasic
13) (secante di x) y = Sec x	**y = 1 / COS (x)**
14) (cosecante di x) y = Cosec x	**y = 1 / SIN (x)**
15) (cotangente di x) y = Cot x	**y = 1 / TAN (x)**

Nella simbologia Qbasic gli argomenti devono essere messi sempre tra parentesi (x).

1.9.3 Le funzioni esponenziali e logaritmiche

Le funzioni logaritmiche riportate in simbologia ordinaria attribuiscono rispettivamente al simbolo " ln " la valenza di logaritmo in base "e" ed al simbolo " log " la valenza di logaritmo in base " 10 " .
La corrispondenza tra le funzioni in simbologia ordinaria e in simbologia Qbasic implementata è data da:

Simbologia ordinaria | Simbologia Qbasic

16) (esponenziale)
$y = e^{x}$ | **y = EXP (x)**

17) (logaritmo naturale di x)
$y = \ln x$ | **y = LOG (x)**

La corrispondenza tra le funzioni in simbologia ordinaria e la simbologia implementabile in Qbasic è la seguente:

Simbologia ordinaria | Simbologia Qbasic

18) (logaritmo in base 10 di x)
$y = \log\ x$ | **y = LOG (x) / LOG (10)**

Nella simbologia Qbasic la variabile x deve essere sempre messa tra parentesi (x)

1.9.4 Le funzioni ciclometriche

Una sola funzione ciclometrica (funzione trigonometrica inversa) è implementata in Qbasic: l'Arcotangente .
La corrispondenza tra la simbologia ordinaria e la simbologia Qbasic è:

Simbologia ordinaria | Simbologia Qbasic

19) (arcotangente di x)
y = Arcotang x | **y = ATN (x)**

La funzione ciclometrica Arcoseno è implementabile come segue:

Simbologia ordinaria | Simbologia Qbasic

20) (Arcoseno di x)
y = Arcosen x | **y = ATN (x / SQR (-x * x + 1))**

Nella simbologia Qbasic gli argomenti devono essere messi sempre tra parentesi (x).

1.9.5 Le funzioni iperboliche
Le funzioni iperboliche sono implementabili in Qbasic mediante il seguente prospetto:

Simbologia ordinaria	Simbologia Qbasic
21) (coseno iperbolico di x) y = ch x	**y = (EXP (x) + EXP (-x)) / 2**
22) (seno iperbolico di x) y = sh x	**y = (EXP (x) - EXP (-x)) / 2**
23) (tangente iperbolica di x) y = th x	**y = (EXP (x) - EXP (-x)) / (EXP (x) + EXP (-x))**

1.9.6 Funzioni speciali
Rientrano nella categoria delle funzioni speciali quelle che non sono comprese nelle tipologie precedenti; queste funzioni sono implementabili in Qbasic:

Simbologia ordinaria	Simbologia Qbasic
24) (seno di x su x) $y = \frac{\text{Sen } x}{x}$	**y = SIN (x) / x**
25) (gaussiana) $y = e^{-a x^2}$	**y = EXP (-a * x ^ 2)**
26) (fattoriale) y = x !	**y = 1*2*3*4* ......* x**

1.9.7 Le sommatorie algebrica e di funzione

1.9.7.1 La funzione sommatoria algebrica
La funzione sommatoria algebrica è un algoritmo di notevole interesse matematico che è facilmente implementabile in Qbasic. Esponiamo la corrispondenza simbolica ordinaria e Qbasic riservandoci il commento del programma in fase di esercitazione:

Simbologia ordinaria Simbologia Qbasic

27) (sommatoria algebrica)

$$y = \sum_{x=p}^{x=n} x$$

```
FOR x = p  TO  n  STEP s
 k = x
y = y + k
NEXT  x
```

1.9.7.2 La funzione sommatoria di funzione

La funzione sommatoria di funzione, che gioca un ruolo fondamentale in numerose applicazioni tecniche e matematiche, prevede il calcolo della somma progressiva dei valori che una funzione f(x) assume al variare di " x ", a passi di "s", dal valore " x = p " al valore " x = n ".
La sommatoria di funzione è implementabile in Qbasic mediante una routine di calcolo che viene qui indicata senza commenti al programma; in fase applicativa la routine sarà commentata:

Simbologia ordinaria Simbologia Qbasic

28) (sommatoria di funzione)

$$y = \sum_{x=p}^{x=n} f(x)$$

```
FOR x = p  TO  n  STEP s
k = f ( x )
y = y + k
NEXT  x
```

1.9.8 Le funzioni della geometria analitica

Sono riportate in questo paragrafo le funzioni della geometria analitica che sono implementabili in Qbasic:

Simbologia ordinaria Simbologia Qbasic

29) (retta)

$Y = m x + n$ **Y = m * x + n**

30) (parabola)

$Y = a x^2 + b x + c$ **Y = a * x ^ 2 + b * x + c**

31) (circonferenza)

$Y = +/- (r^2 - x^2)^{1/2}$

Y = SQR (r ^ 2 - x ^ 2)
Y = - SQR (r ^ 2 - x ^ 2)

32) (iperbole)

$$Y = \frac{1}{x}$$

Y = 1 / x

33) (ellisse)

$Y = +/- (b : a) (a^2 - x^2)^{1/2}$

Y = (b / a) * SQR (a ^ 2 - x ^ 2)
Y = - (b / a) * SQR (a ^ 2 - x ^ 2)

34) (parabola con asse orizzontale)

$Y = +/- (2 p x)^{1/2}$

Y = SQR (2 * p * x)
Y = - SQR (2 * p * x)

1.10 L'impiego delle memorie di calcolo

Nello sviluppo delle routine di programma si impiegano numerose "memorie di calcolo".
L'uso delle memorie di calcolo è fondamentale nella composizione dei programmi, con esse infatti si possono implementare complessi algoritmi matematici e importanti sistemi grafici.
L'uso delle memorie nel contesto della stesura di un programma in Qbasic è automatico ed estremamente semplice:
-Se digitiamo A = .3759 abbiamo automaticamente assegnato alla memoria da noi denominata A il valore di .3759.
-Se digitiamo B1 = A abbiamo imposto che il contenuto della memoria A sia caricato anche nella memoria B1.
-Se si digita K = 0 si azzera il contenuto della memoria K.
-Se digitiamo a(23) = -13.89 abbiamo assegnato alla memoria a(23) il valore di -13.89.
Come si è visto le diverse memorie si possono identificare a piacere con lettere o con combinazioni di lettere e numeri:
Non vi è una limitazione pratica al numero delle memorie che possono essere utilizzate in un programma.
Le memorie di calcolo sono volatili, soltanto la struttura del programma nel quale sono state inserite consente il loro impiego con i valori voluti; se in un programma si è posto ad esempio Q = -3.756 un altro programma che dovesse impiegare la memoria Q troverebbe Q = 0.
Le memorie non possono essere rappresentate da soli numeri, la scritta T632 individua una memoria volatile, alla scritta 632 non è associabile nessuna locazione di memoria.
Le memorie possono indifferentemente contenere dati fissi, tipo C = 72.329, come dati variabili quali, ad esempio, quelli risultanti da un calcolo di una funzione R = f(x), dove nella memoria R transitano, uno dopo l'altro, i valori assunti da f(x).

Il trasferimento dei dati tra memorie può essere in cascata:
F = S
Y = F
E = Y
in questo modo il valore contenuto nella memoria S viene a trovarsi anche nella memoria E.

1.11 Specificazione in merito alla stesura dei programmi nel testo

La stesura dei programmi nel testo non può essere contenuta sempre nell'ambito di una pagina a causa sia delle necessità tipografiche, sia delle dimensioni di alcuni programmi .
Onde evitare al lettore spiacevoli errori di digitazione al P.C. si stabilisce di indicare la prosecuzione di un programma nella pagina successiva con il simbolo & posto all'estremità destra dell'ultima riga di programma o di commento.

1.12 Condizioni di blocco nell'esecuzione di un programma

Un programma, formalmente corretto, può bloccarsi in esecuzione se i dati in esso introdotti creano condizioni anomale.
I casi più frequenti nel calcolo numerico sono denunciati dalla comparsa sullo schermo delle seguenti indicazioni:

Divisione per zero
la scrittura compare, con il blocco del programma, quando il denominatore di una frazione si azzera a seguito dell'introduzione di un dato.

Overflow
la scrittura compare, con il blocco del programma, quando un valore calcolato eccede la capacità numerica specificata .

Chiamata di funzione illegale
la scrittura compare, con il blocco del programma, quando ad esempio si chiede il computo del logaritmo di un numero negativo.

Le scritte menzionate si rimuovono pigiando il tasto **ESC.**

Rimossa la scritta si riavvia il programma bloccato con le quattro seguenti azioni consecutive:
1) si clicca su **File**
2) si clicca su **Apri...**
3) si pigia nuovamente il tasto **ESC**
4) si pigia il tasto **F5** .

CAPITOLO 2
ESERCITAZIONI NUMERICHE DI PROGRAMMAZIONE

Questo capitolo è dedicato ad esercitazioni numeriche da eseguire con il P.C. per imparare la tecnica di programmazione in Qbasic mediante l'uso di tutte le funzioni riportate nei paragrafi dal 1.9.1 al 1.9.8. I paragrafi relativi alle esercitazioni sono intercalati da paragrafi inerenti a consigli e procedure di carattere operativo e di gestione programmi.
I programmi che andremo a compilare saranno di matematica generale senza riferimenti specifici a problematiche tecniche, argomento questo che sarà affrontato più avanti.
Gli esercizi permetteranno le manipolazioni delle funzioni elementari, singolarmente e nelle diverse strutture combinate quali funzioni miste, di più variabili, parametriche, funzioni di funzioni.

Salvo indicazioni diverse tutti gli argomenti che compariranno nelle funzioni trigonometriche saranno sempre espressi in radianti.

Ad eccezione del primo esercizio non si indicherà mai la pressione del tasto (invio) **che deve essere premuto dopo ogni digitazione** di istruzione o valore numerico.
E' utile rimarcare che le istruzioni possono essere digitate con lettere minuscole che, dopo il comando (invio), se digitate correttamente, vengono modificate automaticamente in lettere maiuscole dando così all'operatore la sicurezza di non aver commesso errori di sintassi.
Ad eccezione del primo esercizio non si specificherà più la pressione del tasto F5 indicando soltanto il simbolo F5 per un solo comando e 2F5 per due comandi in successione.
Ad eccezione del primo esercizio non si farà mai riferimento al paragrafo in cui sono riportate le corrispondenze tra funzioni in simbologia ordinaria e funzioni in simbologia Qbasic; sarà soltanto citato il numero d'ordine con il quale sono distinte.

2.1 Esercizio di programmazione n° 1 (funzione aritmetica)

Ci proponiamo di computare la funzione

$$Y = (x)^{1/2}$$

per tre valori della variabile indipendente: x = 7; x = 12; x = 3 .75
allo scopo, vista la corrispondenza simbolica 1) di paragrafo 1.9.1 che riportiamo:

Simbologia ordinaria	Simbologia Qbasic
1) (radice quadrata) $Y = (x)^{1/2}$	**Y = SQR (x)**

compiliamo il programma digitando le seguenti istruzioni:

CLS (invio)

INPUT " x " ; x (invio)

Y = SQR (x) (invio)

PRINT "Y = " ; Y (invio)

terminata la stesura del programma procediamo all'esecuzione:

(premere F5) e digitare il primo valore di x:

x ? 7 (invio)
Y = 2.645751
(premere due volte F5) e digitare il secondo valore di x :
x ? 12 (invio)
Y = 3.464102
(premere due volte F5) e digitare il terzo ed ultimo valore di x:
x ? 3.75 (invio)
Y = 1.936492
(premere F5) per ritornare alla schermata di programma

2.2 Esercizio di programmazione n° 2 (funzioni aritmetiche di due variabili)

Ci proponiamo di computare, contemporaneamente, le tre funzioni di due variabili

$Y = X1 \cdot X2$ $\quad Y = X1 : X2$ $\quad Y = (X1)^{x2}$

che hanno le corrispondenze simboliche in Qbasic rispettivamente nella 3), 4), 2), per le seguenti coppie di valori delle variabili indipendenti:

X1 = 3 X2 = 5
X1 = 2.5 X2 = .97

Per far ciò e necessario attribuire a ciascuna funzione un simbolo diverso in Y; Y1 per la prima, Y2 per la seconda, Y3 per la terza e compilare il programma come segue:

```
CLS

INPUT "X1" ; X1

INPUT "X2" ; X2

Y1 = X1 * X2

Y2 = X1 / X2

Y3 = X1 ^ X2

PRINT"Y1=" ; Y1

PRINT"Y2=" ; Y2

PRINT"Y3=" ; Y3
```

F5

X1 ? 3
X2 ? 5
Y1 = 15
Y2 = .6
Y3 = 243

2F5

X1 ? 2.5
X2 ? .97

Y1 = 2.425
Y2 = 2.57732
Y3 = 2.432214

F5

2.3 Osservazioni in merito alla precisione di calcolo

Nei due esercizi svolti in precedenza abbiamo ottenuto risultati espressi con sei cifre decimali; se si desidera ottenere una precisione maggiore (doppia precisione) si deve posporre alla variabile dipendente il simbolo # come indicato: Y# .
Nel paragrafo seguente si applica questa nuova simbologia nel compilare uno dei programmi oggetto dell'esercizio n° 3.

2.4 Esercizio di programmazione n° 3 (funzione trigonometrica elementare)

Eseguiamo il calcolo della funzione Y = Sen x per il valore di x = .5 (radianti), utilizzando la corrispondenza simbolica 10) e scriviamo:

```
CLS

INPUT "x" ; x

Y = SIN ( x )

PRINT " Y = " ; Y
```

F5

x ? .5
Y = .4794255

F5

Il valore della variabile dipendente che abbiamo ottenuto è a singola precisione. Se si vuole sviluppare il calcolo in doppia precisione si deve ricompilare il programma in base a quanto indicato nel paragrafo 2.3:

```
CLS

INPUT"x" ; x

Y# = SIN ( x )

PRINT "Y" ; Y#
```

F5

x ? .5
Y = .479425538604203

F5

2.5 L'impiego delle costanti

In alcuni casi di compilazione dei programmi devono essere introdotte più costanti uguali; per semplificare questa procedura è sufficiente introdurre la costante una sola volta mediante l'uguaglianza con una lettera, a piacere, e richiamare poi tale lettera nell'ambito delle espressioni matematiche al posto della costante.

Se ad esempio la costante è π prima delle istruzioni che la riportano scriveremo, come fosse una istruzione, l'uguaglianza: p = 3.141592654

2.6 Esercizio di programmazione n° 4 (funzione trigonometrica combinata)

Prima di eseguire l'esercizio è opportuno osservare che con questo paragrafo si introducono le funzioni composte, funzioni che sono il risultato di diverse operazioni, aritmetiche ed analitiche. Questo tipo di funzioni sono utilizzate spesso nel testo, per tale ragione si deve spiegare, una volta per tutte, qual è il criterio che richiama le diverse corrispondenze simboliche.
Se la funzione da computare è ad esempio:

$$Y = \text{Sen } x \text{ Cos } x + \text{Tang } x$$

troviamo nell'ordine: la funzione Sen x alla quale è associata la corrispondenza 10) **SIN (x)**, il prodotto tra Sen x e Cos x al quale è associata la corrispondenza 3) *, la funzione Cos x alla quale è associata la corrispondenza 11) **COS(x)**, la somma del prodotto (Sen x Cos x) con Tang x alla quale è associata la corrispondenza 5) +, ed infine la funzione Tang x alla quale è associata la corrispondenza 12) **TAN (x)** . Questo modo di ragionare sarà nel prosieguo sintetizzato mediante la sola scrittura dei numeri relativi alle corrispondenze, nella sequenza dipendente dal tipo di combinazione. Dopo questa premessa computiamo la funzione già esaminata:

$$Y = \text{Sen } x \text{ Cos } x + \text{Tang } x$$

per due valori della variabile indipendente $x = 3.5°$ e $x = 37°$ sessagesimali.
In questo caso dobbiamo trasformare i valori sessagesimali in radianti, così come abbiamo mostrato all'inizio del paragrafo 1.9.2, impiegando la costante moltiplicativa k = .017453293.
Il programma si compone, nell'ordine, in base alle corrispondenze simboliche 10), 3), 11), 5), 12) e, per semplificarne la compilazione, seguendo le indicazioni del paragrafo 2.5.

```
CLS

INPUT " x " ; x

k = .017453293

Y = SIN ( x * k ) * COS ( x * k ) + TAN ( x * k )

PRINT " Y = " ; Y
```

```
                    F5
x ? 3.5
Y = .1220973
                    2F5
x ? 37
Y = 1.234185

                    F5
```

2.7 Osservazioni in merito all'istruzione CLS

L'istruzione CLS, posta in testa al programma allo scopo di pulire lo schermo video, può essere omessa se si desiderano raccogliere in una sola schermata più elaborazioni numeriche di uno stesso programma. Naturalmente, in questo caso, è necessario prestare attenzione a non confondere risultati di elaborazioni precedenti con risultati dell'ultima elaborazione.

2.8 Esercizio di programmazione n° 5 (funzione trigonometrica elementare)

Computiamo la funzione Y = Sec x per quattro valori di x .2; .4; .6; .8 radianti, raccogliendo tutte e quattro le operazioni di INPUT ed i quattro valori di Y in una sola schermata.
Per evitare di trovare sullo schermo dati calcolati in precedenza dobbiamo avere l'accortezza di digitare sulla schermata CLS e premere 2F5, quindi:
Secondo quanto indicato al paragrafo 2.7 omettiamo l'istruzione CLS ed in base alla corrispondenza simbolica 13) scriviamo:

```
INPUT " x " ; x

Y = 1/ COS ( x )

PRINT " Y = " ; Y
```

```
                F5
x ? .2
Y =  1.020339
                2F5
x ? .4
Y =  1.085704
                2F5
x ? .6
Y =  1.211628
                2F5
x ? .8
Y =  1.435324
                F5
```

2.9 Modalità di titolazione e memorizzazione di un programma compilato

Dopo aver compilato un programma, ed averne controllato la correttezza mediante valutazione dei valori assunti dalla variabile indipendente, lo si può titolare e memorizzare in modo che risulti nella memoria del P.C. come documento di lavoro. La procedura di titolazione e memorizzazione è molto semplice e si avvale della lista di operazioni di gestione, già citata al paragrafo 1.1, che si evidenzia cliccando col mouse sulla scritta **File**.
A seguito del comando del mouse su File compare, tra le altre, la scritta **salva con nome...**, cliccando poi su questa scritta viene presentata sul video una particolare maschera in cui, dentro un apposito riquadro, è riportata la scritta: **Nome del file**, a questo punto si digita il nome che vogliamo assegnare al programma seguito da un punto e dalla dizione BAS come nell'esempio che segue:

Nome del file PROVA.BAS (invio)

dopo il comando invio compare in alto sullo schermo, al posto della scritta "senza titolo" la nuova dicitura " PROVA.BAS ".
Nell'eseguire questa operazione si deve tenere presente che il nome che vogliamo assegnare al programma non deve avere più di 8 caratteri; se digitiamo più di 8 caratteri e premiamo (invio) compare un riquadro nel quale è indicato " nome di file errato", per correggere dobbiamo premere il tasto ESC e modificare entro il numero di caratteri stabilito.
Una volta che il programma è titolato, sia che si proceda ad altro lavoro, sia che si ritenga di sospendere, si può memorizzarlo cliccando su **File** e di seguito su **Salva**.

2.10 Esercizio di programmazione n° 6 (funzione parametrica composta)

Costruiamo il programma per il calcolo della seguente funzione parametrica composta:

$$Y = \text{Cosec}\, x + A \;\text{Cot}\, x$$

da computare per due valori del parametro A e per quattro valori della variabile indipendente .
Per A 1.5 e 3, per x .1; .2; .3; .4 .
E' utile, nel compilare il programma, commentare le diverse istruzioni per ricordare la sequenza degli eventi richiesti .
Digiteremo pertanto in base alle corrispondenze simboliche 14), 5), 3) e15):

```
CLS ' pulizia schermo

INPUT " A " ; A ' richiesta ingresso valore parametro

INPUT " x " ; x ' richiesta ingresso variabile indipendente

Y = 1/ SIN ( x ) + A * ( 1/ TAN ( x ) ) ' calcolo funzione parametrica

PRINT " Y = " ; Y ' comando visualizzazione dati variabile dipendente
```

F5

```
A ? 1.5
x ? .1
Y = 24.96665
```

2F5

```
A ? 1.5
x ? .2
Y = 12.43322
```

2F5

```
A ? 1.5
x ? .3
Y = 8.232955
```

2F5

```
A ? 1.5
x ? .4
Y = 6.115766
```

2F5

```
A ? 3
x ? .1
Y =  39.91662
```

2F5

```
A ? 3
x ? .2
Y = 19.83295
```

2F5

```
A ? 3
x ? .3
Y = 13.08205
```

2F5

```
A ? 3
x ? .4
Y = 9.6636
```

F5

2.11 Sistema automatico per il calcolo di funzioni a campo fisso

L'esercizio del paragrafo precedente ha richiesto una procedura lunga e ripetitiva per ottenere i risultati voluti; infatti il valore del parametro A è stato digitato 8 volte e altre 8 volte sono stati digitati i valori di x. Si può comprendere quanto lavoro richiederebbe la computazione con più parametri estesa a un maggior numero dei valori della variabile indipendente.
Per semplificare la computazione delle funzioni viene in aiuto una semplice routine di programma che prevede, una volta fissato a priori il campo di escursione e l'entità degli incrementi della variabile indipendente, il computo automatico della Y = f (x) in tutto il campo stabilito.
Mostriamo la routine, completa di tutte le istruzioni necessarie per l'applicazione immediata, digitata e commentata per una generica funzione espressa simbolicamente.

```
INPUT " K" ; K  ' richiesta di ingresso del parametro

FOR  x = m  TO  n  STEP s  ' l'istruzione impone che la variabile indipendente (x) inizi il
                           ' calcolo assumendo il valore (m) , quando l'esecuzione del programma giunge
                           ' all 'istruzione NEXT, il programma ritorna all'istruzione FOR che incrementa x
                           ' del valore (s) . Il ciclo si ripete con incrementi di (s) per arrestarsi
                           ' quando il valore di x ha raggiunto il valore (n)

Y =  f ( x ; K )  ' funzione parametrica simbolica da computare

PRINT "Y = " ; Y  ' comando visualizzazione dati variabile dipendente

NEXT x  ' comanda il programma al ritorno automatico all 'istruzione FOR
```

L'applicazione del programma esposto trova riscontro pratico nel paragrafo seguente.

2.12 Esercizio di programmazione n° 7 (funzione parametrica esponenziale)

Visto il sistema automatico per il computo delle funzioni applichiamolo alla funzione parametrica esponenziale:

$$Y = e^{(kx)}$$

che ha nelle 16) e 3) le corrispondenze simboliche in Qbasic.
L'esercizio prevede che il parametro k assuma due valori: .3; .5 e che la variabile indipendente assuma dieci valori; da x = 3 a x = 21 ad intervalli di 2 unità .

Compiliamo il programma commentandolo:

```
CLS  ' pulizia dello schermo

INPUT " k" ; k  ' richiesta di ingresso del parametro

FOR  x = 3  TO  21  STEP 2  ' impone che la variabile indipendente (x) inizi il calcolo assumendo il
                            ' valore (3) , quando l'esecuzione del programma giunge alla istruzione
                            ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
                            ' valore (2) . Il ciclo si ripete con incrementi di (2) per arrestarsi
                            ' quando il valore di x ha raggiunto il valore (21)

Y = EXP ( k * x )  ' funzione parametrica da computare
```

&

PRINT "Y = " ; Y ' comando visualizzazione dati variabile dipendente

NEXT x ' comanda il programma al ritorno automatico alla istruzione **FOR** fino al raggiungimento ' dell'ultimo valore previsto per la variabile indipendente.

```
                F5
k ? .3
Y = 2.459603
Y = 4.481689
Y = 8.16617
Y = 14.87973
Y = 27.11264
Y = 49.40246
Y = 90.01715
Y = 164.0219
Y= 298.8675
Y = 544.572
                2F5
k ? .5
Y = 4.481689
Y =12.18249
Y =33.11545
Y =90.01713
Y =244.6919
Y =665.1417
Y =1808.042
Y =4914.769
Y=13359.73
Y =36315.5
                F5
```

2.13 Sistema automatico per il calcolo di funzioni a campo variabile

La routine di calcolo illustrata nel paragrafo 2.11 può essere estesa per la computazione di funzioni nell'ambito di un campo variabile, in escursione ed entità degli incrementi, della x.
Mostriamo la routine, completa di tutte le istruzioni necessarie per l'applicazione immediata, digitata e commentata per una generica funzione espressa simbolicamente.

INPUT " K" ; K ' richiesta di ingresso del parametro

INPUT " m " ; m ' richiesta di ingresso dell'estremo inferiore del campo di variabilità di x

INPUT " n " ; n ' richiesta di ingresso dell'estremo superiore del campo di variabilità di x

INPUT " s " ; s ' richiesta di ingresso dell'entità dell'incremento di x nell'ambito del campo

FOR x = m TO n STEP s ' impone che la variabile indipendente (x) inizi il calcolo assumendo il ' valore (m) , quando l'esecuzione del programma giunge alla istruzione ' **NEXT,** il programma ritorna all'istruzione **FOR** che incrementa x del ' valore (s) . Il ciclo si ripete con incrementi di (s) per arrestarsi ' quando il valore di x ha raggiunto il valore (n)

Y = f (x ; K) ' funzione parametrica da computare

PRINT "Y = " ; Y ' comando visualizzazione dati variabile dipendente

NEXT x ' comanda il programma al ritorno automatico alla istruzione **FOR**

L'applicazione del programma esposto trova riscontro pratico nel paragrafo seguente.

2.14 Esercizio di programmazione n° 8 (funzione logaritmica composta)

Per esercitarci sulla routine di calcolo automatico a campo variabile, descritta nel paragrafo precedente, computiamo la seguente funzione logaritmica composta:

$$Y = \ln x \cdot \log x$$

Prima di procedere alla stesura del programma e necessario fare una importante osservazione relativa al campo di variabilità della x; la funzione che ci accingiamo a computare è del tipo logaritmico e pertanto non può essere calcolata né per x = 0, valore per il quale è infinitamente grande, né per valori negativi di x che sono estranei al campo di variabilità di questo tipo di funzione. Per evitare quindi che il programma porti ad una condizione di errore, denunciata dalla indicazione **"CHIAMATA DI FUNZIONE NON VALIDA"**, il valore di (x) deve essere, in questo esempio di calcolo, sempre maggiore di zero; **x > 0** .
Ciò premesso stabiliamo per il nostro esercizio il campo di variabilità nei seguenti limiti:
m = .1; n = 1 con incremento di x pari ad s = .2
In base alle corrispondenze simboliche 17); 3); 18) ed al programma del paragrafo 2.13, nel quale omettiamo la prima istruzione non essendo questo esercizio su di una funzione parametrica, compiliamo il nuovo programma:

```
CLS ' pulizia dello schermo

INPUT " m " ; m ' richiesta di ingresso dell'estremo inferiore del campo di variabilità di x

INPUT " n " ; n ' richiesta di ingresso dell'estremo superiore del campo di variabilità di x

INPUT " s " ; s ' richiesta di ingresso dell'entità dell'incremento di x nell'ambito del campo

FOR x = m TO n STEP s ' impone che la variabile indipendente (x) inizi il calcolo assumendo il
                      ' valore (m) , quando l'esecuzione del programma giunge alla istruzione
                      ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
                      ' valore (s) . Il ciclo si ripete con incrementi di (s) per arrestarsi
                      ' quando il valore di x ha raggiunto il valore (n)

Y = LOG ( x ) * LOG ( x ) / LOG ( 10 ) ' funzione logaritmica composta da computare

PRINT "Y = " ; Y ' comando visualizzazione dati variabile dipendente

NEXT x ' comanda il programma al ritorno automatico alla istruzione FOR
```

F5

```
m ? .1
n ? 1
s ? .2
Y = 2.302585
Y = .6295317
Y = .2086581
Y = 5.524965E-02
Y = 4.821035E-03
```

F5

I risultati che abbiamo ottenuto evidenziano, per la prima volta in un esercizio, le **notazioni scientifiche E-02**; **E-03**, esse indicano che le cifre poste alla loro sinistra devono essere divise rispettivamente per 100 e per 1000; la prima cifra vale 0.05524965 e la seconda vale 0.004821035.

In generale:

la notazione E - 0 n implica la divisione del numero alla sua sinistra per 10^n

la notazione E + 0 n implica la moltiplicazione del numero alla sua sinistra per 10^n

La semplicità e la rapidità di questo metodo di calcolo si commentano da sole, il computo della funzione può essere ripetuto a piacimento per qualsiasi campo di variabilità della x con l'incremento che può essere scelto in base alle necessità di definizione del calcolo; infatti più piccolo sarà il valore assegnato ad (s) maggiore sarà il numero dei punti calcolati, nell'ambito del campo imposto. Visti i risultati che abbiamo ottenuto con il sistema automatico di calcolo, gran parte degli esercizi a seguire saranno svolti secondo questa metodologia.

2.15 Come richiamare in video un programma precedentemente memorizzato

Se si vuole richiamare in video un programma, già memorizzato secondo quanto indicato nel paragrafo 2.9, si procede come segue:
Si clicca con il mouse su **File** e di seguito su **Apri......** ; si ha la comparsa di un riquadro con la scritta **Nome del file:**
Si digita il nome del programma, ad esempio PROVA.BAS, e si preme (invio); si ha immediatamente sul video la comparsa della schermata del programma richiesto.

2.16 Come documentare un programma mediante stampante

Se si desidera documentare un programma mediante stampante, in modo da poterlo consultare in qualsiasi momento senza la necessità di operare con il P.C., si utilizza la sua presentazione sul video: o con la procedura indicata nel paragrafo precedente, o dopo averne ultimato la compilazione, lo si può quindi stampare semplicemente come sotto indicato:
Si clicca con il mouse su **File** e di seguito su **Stampa ....**; si ha la comparsa di un riquadro sul video con le scritte:

( ) **Solo testo selezionato**
( ) **Finestra corrente**
( ) **Programma intero**

si clicca con il mouse all'interno dell' ultima parentesi e si preme (invio); la stampante si mette in azione e tutto è fatto. La stampa che si ottiene è detta **listato del programma.**
Per una più completa documentazione di un programma è opportuno allegare al suo listato la stampa di un computo numerico relativo ad una sua applicazione, da ottenere con le modalità indicate nel paragrafo 1.4.2.2 o 1.4.2.3.

2.17 Esercizio di programmazione n° 9 (funzione di funzione)

Per esercitarci con le funzioni di funzioni proponiamoci il computo della funzione ciclometrica seguente:

Y = Arcotang (t)

dove (t) è a sua volta dipendente da un'altra funzione del tipo:

$$t = e^{x}$$

dove x è la sola variabile indipendente . Le due funzioni possono essere rappresentate con una sola espressione sostituendo alla lettera (t) della prima espressione l'esponenziale della seconda:

$$Y = \text{Arcotang} (e^{x})$$

Assumendo come valori del campo di variabilità della x m =.1 n = 8.1 con s = 2, mediante le corrispondenze simboliche 19) e 16) possiamo compilare il programma:

CLS ' pulizia dello schermo

INPUT " m " ; m ' richiesta di ingresso dell'estremo inferiore del campo di variabilità di x

INPUT " n " ; n ' richiesta di ingresso dell'estremo superiore del campo di variabilità di x

INPUT " s " ; s ' richiesta di ingresso dell'entità dell'incremento di x nell'ambito del campo

FOR x = m TO n STEP s ' impone che la variabile indipendente (x) inizi il calcolo assumendo il
' valore (m) , quando l'esecuzione del programma giunge alla istruzione
' **NEXT,** il programma ritorna all'istruzione **FOR** che incrementa x del
' valore (s) . Il ciclo si ripete con incrementi di (s) per arrestarsi
' quando il valore di x ha raggiunto il valore (n)

Y = ATN (EXP (x)) ' funzione di funzione da computare

PRINT "Y = " ; Y ' comando visualizzazione dati variabile dipendente

NEXT x ' comanda il programma al ritorno automatico alla istruzione **FOR**

```
          F5
m ? .1
n ? 8.1
s ? 2
Y = .835315
Y= 1.448947
Y=1.554225
Y=1.568553
Y=1.570493
          F5
```

2.18 Esercizio di programmazione n° 10 (funzione di funzione)

Dato il diffuso impiego delle funzioni di funzioni proponiamo un secondo esercizio allo scopo di abituare il lettore alla manipolazione spedita di questo importante algoritmo.
Consideriamo la funzione ciclometrica:

$$Y = \text{Arcsen } t$$

nella quale la variabile (t) dipende a sua volta dalla funzione

$$t = b^{3}$$

nella quale la variabile (b) dipende infine da x secondo la funzione

$$b = \text{Cos } x$$

Le tre formule possono essere rappresentate da un solo algoritmo mediante le sostituzioni d'uso:

$$Y = \text{Arcsen } (\text{Cos } x)^3$$

La formula ottenuta ha il pregio di sintetizzare un considerevole numero di operazioni ma presenta difficoltà di implementazione in Qbasic se non si ha un poco di esperienza.
E' pertanto consigliabile, almeno in prima battuta, procedere alla compilazione del programma introducendo le tre funzioni separatamente mediante un metodo che sarà illustrato nei commenti delle diverse istruzioni.
Il metodo che impiegheremo, se in questo esercizio viene in aiuto al principiante, è in effetti uno strumento indispensabile quando ci si trovi a dover implementare, in routine di programma, funzioni di funzioni che sono proposte mediante un unico algoritmo molto complicato, tale da non essere trasformabile direttamente mediante le corrispondenze simboliche; è in questi casi che si rende necessario suddividere la funzione principale complicata in più funzioni semplici in modo da consentirne una agevole implementazione similmente a quanto ci accingiamo ora a fare.
Con l'aiuto delle corrispondenze simboliche 11); 2); 20) per m = .5; n = 1; s = .1 si compila il seguente programma:

```
CLS ' pulizia dello schermo

INPUT " m " ; m ' richiesta di ingresso dell'estremo inferiore del campo di variabilità di x

INPUT " n " ; n ' richiesta di ingresso dell'estremo superiore del campo di variabilità di x

INPUT " s " ; s ' richiesta di ingresso dell'entità dell'incremento di x nell'ambito del campo

FOR x = m TO n STEP s ' impone che la variabile indipendente (x) inizi il calcolo assumendo il
                      ' valore (m) , quando l'esecuzione del programma giunge alla istruzione
                      ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
                      ' valore (s) . Il ciclo si ripete con incrementi di (s) per arrestarsi
                      ' quando il valore di x ha raggiunto il valore (n)

b = COS ( x ) ' si computa inizialmente la funzione legata direttamente alla variabile indipendente x
              ' che nominiamo "prima"

t = b ^ 3     ' si computa di seguito la funzione legata direttamente alla "prima"
              ' che denominiamo "seconda"

Y = ATN ( t / SQR ( - t * t + 1 ) ) ' si computa infine la "terza" funzione legata direttamente alla
                                    ' seconda
PRINT "Y = " ; Y ' comando visualizzazione dati "terza" variabile dipendente

NEXT x ' comanda il programma al ritorno automatico alla istruzione FOR
```

```
                F5
m ? = .5
n ? = 1
s ? = .1
```

Y=.7421461
Y=.5970451
Y=.4638785
Y=.3449839
Y=.2425609

F5

Non è superfluo ricordare che il programma è compilato in modo che il computo della funzione di funzione possa essere ripetuto per qualsivoglia campo di variabilità della variabile indipendente x, essendo flessibile la determinazione dei valori di (m) ed (n), e che nell'ambito del campo di variabilità di x si possono calcolare quanti valori della variabile indipendente Y siano necessari per ottenere una accurata analisi della funzione dipendente; ciò è infatti possibile mediante la definizione dell'incremento (s) .

2.19 Sistema automatico per il calcolo delle funzioni di più variabili in campi diversi

La routine di calcolo mostrata nel paragrafo 2.18 può essere estesa per la computazione di funzioni di più variabili nell'ambito di campi diversi, in escursione ed entità degli incrementi, delle variabili indipendenti.
Per semplicità di esposizione esamineremo il caso di funzioni a 2 variabili, l'estensione ad un maggior numero di queste si potrà poi ricavare da quanto verrà spiegato.
Ricordiamo la notazione simbolica esplicita di una funzione di due variabili:

$$Y = f(x1 ; x2)$$

per questa funzione il calcolo si sviluppa assegnando a x1 il primo valore voluto ed andando poi a computare i valori assunti dalla Y per tutti gli altri valori di x2, si ripete l'operazione per il secondo valore assegnato a x1 per tutti i valori che deve assumere x2, si procede in questo modo fino all'ultimo valore di x1 al quale associare sempre tutti i valori che deve assumere x2; pertanto il numero dei valori assunti dalla variabile dipendente è pari al prodotto del numero dei valori assunti da x1 ed i valori assunti da x2.
Se ad esempio per x1 si stabiliscono 10 valori e per x2 7 valori la variabile dipendente assumerà 70 valori.
Sulla base di quanto detto mostriamo la routine di calcolo commentata per una funzione a 2 variabili

```
INPUT " m1 " ; m1  ' richiesta di ingresso dell'estremo inferiore del campo di variabilità di x1

INPUT " n1 " ; n1  ' richiesta di ingresso dell'estremo superiore del campo di variabilità di x1

INPUT " s1 " ; s1  ' richiesta di ingresso dell'entità dell'incremento di x1 nell'ambito del campo

INPUT " m2 " ; m2  ' richiesta di ingresso dell'estremo inferiore del campo di variabilità di x2

INPUT " n2 " ; n2  ' richiesta di ingresso dell'estremo superiore del campo di variabilità di x2

INPUT " s2 " ; s2  ' richiesta di ingresso dell'entità dell'incremento di x2 nell'ambito del campo

FOR x1 = m1 TO n1 STEP s1 ' impone che la variabile indipendente (x1) inizi il calcolo
                          ' assumendo il valore (m1) , quando l'esecuzione del programma
                          ' giunge alla istruzione NEXT x1, il programma ritorna all'istruzione
                          ' FOR che incrementa x1                                          &
```

```
FOR  x2 = m2  TO  n2  STEP s2  ' impone che  la  variabile indipendente (x2) inizi il calcolo
                               ' assumendo il valore (m2) , quando l'esecuzione del programma
                               ' giunge alla istruzione NEXT x2, il programma ritorna all'istruzione
                               ' FOR  che incrementa x2

Y =  f ( x1 ; x2 )  ' funzione  di due variabili da computare

PRINT "Y = " ;  Y  ' comando visualizzazione dati variabile dipendente

NEXT x2  ' comanda il programma al ritorno automatico alla istruzione FOR x2 = ...

NEXT x1  ' comanda il programma al ritorno automatico alla istruzione FOR x1 = ...
```

Come si vede la routine si sviluppa come abbiamo premesso; viene assegnato ad x1 il primo valore, e di seguito tutti i valori di x2 prima che x1 subisca il primo incremento, quando tutti i valori di x2 sono stati introdotti x1 si incrementa del primo valore pari ad s1 e nuovamente vengono assegnati di seguito tutti i valori di x2; il processo continua fino a quando x1 a seguito dei successivi incrementi raggiunge il valore di n1.

2.19.1 Esercizio di programmazione n° 11 (funzione di due variabili in automatico)

Questo esercizio si basa sulla routine automatica per il computo delle funzioni di due variabili spiegata nel paragrafo 2.19, viene computata la seguente funzione trigonometrica composta:

$$Y = \text{Sen } x1 + \text{Cos } x2$$

in due campi di variabilità così definiti:

per x1   da m1 = 0 a   n1 = .2 radianti con s1 = .1
per x2   da m2 = 0 a   n2 = .6 radianti con s2 = .2

Il primo campo comprende 3 valori della variabile indipendente x1, il secondo campo comprende 4 valori della variabile indipendente x2, per conseguenza la variabile dipendente assumerà 12 valori.
In base alle corrispondenze simboliche 10); 5); 11) si compila il seguente programma:

```
CLS  ' pulizia schermo

INPUT " m1 " ;  m1  ' richiesta di ingresso  dell'estremo inferiore del campo di variabilità di x1

INPUT " n1 " ;  n1   ' richiesta di ingresso  dell'estremo superiore  del campo di variabilità di x1

INPUT " s1 " ;  s1   ' richiesta di ingresso  dell'entità dell'incremento di x1  nell'ambito del campo

INPUT " m2 " ;  m2  ' richiesta di ingresso  dell'estremo inferiore del campo di variabilità di x2

INPUT " n2 " ;  n2   ' richiesta di ingresso  dell'estremo superiore  del campo di variabilità di x2

INPUT " s2 " ;  s2   ' richiesta di ingresso  dell'entità dell'incremento di x2 nell'ambito del campo

FOR  x1 = m1  TO  n1  STEP s1  ' impone che  la  variabile indipendente (x1) inizi il calcolo
                               ' assumendo il valore (m1) , quando l'esecuzione del programma
                               ' giunge alla istruzione NEXT x1, il programma ritorna all'istruzione
                               ' FOR  che incrementa x1
FOR  x2 = m2  TO  n2  STEP  s2  ' impone che  la  variabile indipendente (x2) inizi il calcolo          &
```

```
                                     ' assumendo il valore (m2) , quando l'esecuzione del programma
                                     ' giunge alla istruzione NEXT x2, il programma ritorna all'istruzione
                                     ' FOR  che incrementa x2

Y =  SIN ( x1 )  + COS ( x2 )  ' funzione  di due variabili da computare

PRINT "Y = " ;  Y   ' comando visualizzazione dati variabile dipendente

NEXT x2  ' comanda il programma al ritorno automatico alla istruzione FOR x2 =

NEXT x1  ' comanda il programma al ritorno automatico alla istruzione FOR x1 =
```

```
            F5
m1 ? 0
n1  ? .2
s1  ? .1
m2 ? 0
n2  ? .6
s2  ? .2
Y=1
Y=.9800666
Y=.921061
Y=.8253356
Y=1.099833
Y=1.0799
Y=1.020894
Y=.925169
Y=1.198669
Y=1.178736
Y=1.11973
Y=1.024005
            F5
```

2.20 Esercizio di programmazione n° 12 (funzione iperbolica)

Gli esercizi di programmazione svolti finora hanno interessato tutte le funzioni elementari, 20) compresa; proseguiamo il lavoro computando la funzione iperbolica 23), tangente iperbolica, dato che questa esprime tanto il seno 21) quanto il coseno iperbolico 22) in quanto è data dal loro rapporto.

La tangente iperbolica è espressa in modo convenzionale da:

$$Y = \text{th } x$$

ci proponiamo di computarla nel campo di variabilità della x compreso tra .1 e .6 con incrementi di .1.

Compiliamo il programma con l'ausilio della corrispondenza simbolica 23) e con il sistema automatico a campo fisso illustrato nel paragrafo 2.11.

```
CLS  ' pulizia dello schermo

FOR  x = .1  TO  .6  STEP .1  ' impone che la variabile indipendente (x) inizi il calcolo assumendo
                              ' il valore ( .1 ) , quando l'esecuzione del programma giunge alla
                              ' istruzione  NEXT, il programma ritorna all'istruzione FOR che
                              ' incrementa x del valore (.1 ) . Il ciclo si ripete con incrementi di (.1)
                              ' per arrestarsi quando il valore di x ha raggiunto il valore ( .6 )

Y =  ( EXP ( x ) - EXP ( -x  ) ) / ( EXP ( x ) + EXP ( -x  ) ) ' funzione iperbolica da computare          &
```

PRINT "Y = " ; Y ' comando visualizzazione dati variabile dipendente

NEXT x ' comanda il programma al ritorno automatico alla istruzione **FOR** fino al raggiungimento
' dell'ultimo valore previsto per la variabile indipendente.

F5

Y =.099668
Y=.1973753
Y=.2913126
Y=.379949
Y=.4621172
Y=.5370496

F5

2.21 Esercizio di programmazione n° 13 (funzione Sen x / x)

La prima delle funzioni speciali che andiamo a computare è la funzione fratta

$$Y = \frac{\text{Sen } x}{x}$$

questa funzione ha una particolarità; è indeterminata per x = 0 dato che per questo valore della variabile indipendente tanto il numeratore che il denominatore sono nulli. E' chiaro che se noi tentiamo il computo della funzione per x = 0 il Qbasic blocca l'esecuzione del calcolo indicando **"Chiamata di funzione illegale"** mentre sappiamo che il valore della funzione, per x che tende a 0, tende ad 1. E' pertanto indispensabile che il campo di variabilità della x non comprenda il valore 0; se si ha necessità di operare per tale valore lo si deve sostituire con un numero molto piccolo quale ad esempio .000001.
Ciò premesso possiamo compilare il programma in base alla corrispondenza simbolica 24) ed al sistema automatico a campo variabile per m = .000001 n = 2 s = .2 .

CLS 'pulizia dello schermo

INPUT " m " ; m ' richiesta di ingresso dell'estremo inferiore del campo di variabilità di x

INPUT " n " ; n ' richiesta di ingresso dell'estremo superiore del campo di variabilità di x

INPUT " s " ; s ' richiesta di ingresso dell'entità dell'incremento di x nell'ambito del campo

FOR x = m TO n STEP s ' impone che la variabile indipendente (x) inizi il calcolo assumendo il
' valore (m) , quando l'esecuzione del programma giunge alla istruzione
' **NEXT,** il programma ritorna all'istruzione **FOR** che incrementa x del
' valore (s) . Il ciclo si ripete con incrementi di (s) per arrestarsi
' quando il valore di x ha raggiunto il valore (n)
Y = SIN (x) / x ' funzione da computare

PRINT "Y = " ; Y ' comando visualizzazione dati variabile dipendente

NEXT x ' comanda il programma al ritorno automatico alla istruzione **FOR**

F5

m ? .000001
n ? 2
s ? .2
Y=1

Y=.9933466
Y=.9735457
Y=.9410706
Y=.8966949
Y=.8414707
Y=.7766989
Y=.7038923
Y=.624733
Y= .541026

F5

2.21.1 Osservazioni sulle funzioni fratte

Nell'esercizio precedente abbiamo preso delle precauzioni nel calcolo della funzione Sen x /x onde evitarne l'indeterminazione. In generale nel computare qualsiasi funzione fratta, dotata cioè di denominatore funzione della variabile indipendente, è indispensabile calcolare gli eventuali zeri del denominatore per escluderli opportunamente dal campo di variabilità, sia per evitare forme indeterminate, sia per evitare forme di infinito.

2.22 Esercizio di programmazione n° 14 (la funzione gaussiana)

La funzione gaussiana, che gioca un ruolo importante nel campo dell'analisi dei segnali, è una particolare funzione di funzione, esponenziale e parametrica, in cui l'esponente varia con il prodotto del parametro con il quadrato della variabile indipendente secondo l'espressione:

$$y = e^{-a x^2}$$

Nonostante la complessità della dipendenza di y dalla variabile indipendente la corrispondenza simbolica in Qbasic è molto semplice ed è espressa mediante la 25); corrispondenza secondo la quale ne eseguiamo il computo per un campo di variabilità della x compreso tra 1 e 5 con incrementi a passi unitari e parametro a = .01.

```
CLS  ' pulizia dello schermo

a = .01  ' valore del parametro

FOR  x = 1 TO  5  STEP 1  ' impone che la variabile indipendente (x) inizi il calcolo assumendo il
                          ' valore (1) , quando l'esecuzione del programma giunge alla istruzione
                          ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
                          ' valore (1) . Il ciclo si ripete con incrementi di (1) per arrestarsi
                          ' quando il valore di x ha raggiunto il valore (5)

Y = EXP ( - a  *  x ^ 2 )   ' funzione gaussiana da computare

PRINT "Y = " ;  Y   ' comando visualizzazione dati variabile dipendente

NEXT x   ' rimanda all'istruzione  FOR  per un successivo incremento di x
```

F5

Y=.9900498
Y=.9607894
Y=.9139312
Y=.8521438
Y=.7788008

F5

E' opportuno osservare che i metodi di computo automatico a campo fisso quando, come in questo caso, hanno il valore dell'incremento (s) = 1 possono essere compilati omettendo la notazione **STEP 1**.

2.23 Esercizio di programmazione n° 15 (il fattoriale)

L'esercizio relativo alla funzione fattoriale, così come lo andiamo ora a sviluppare, è molto semplice e di poca eleganza formale; un metodo automatico è applicabile ma di scarso interesse pratico. La funzione fattoriale è espressa da:

$$Y = x!$$

procediamo al computo di un unico valore per x = 6 mediante la corrispondenza simbolica 26):

```
CLS

Y = 1 * 2 * 3 * 4 * 5 * 6

PRINT "Y = " ; Y

            F5
Y = 720
            F5
```

2.24 Esercizio di programmazione n° 16 (sommatoria algebrica)

La sommatoria algebrica ricorre sovente nei problemi tecnici e matematici, la funzione che la caratterizza è la seguente

$$Y = \sum_{x=p}^{x=n} x$$

essa ha lo scopo di eseguire il computo della somma di un certo numero di valori assunti dalla variabile indipendente x; se ad esempio p = 5 e n = 9 e l'incremento di x è unitario, per computare Y dobbiamo sommare i valori 5 + 6 + 7 + 8 + 9 ottenendo Y = 35. L'operazione di per se è banale ma il concetto di sommatoria è generalmente applicato a funzioni complicate e pertanto il suo computo diventa impegnativo.
Se vogliamo ora implementare la funzione in un programma in Qbasic e computarla per p =1, n = 5 con incrementi a passi di s =.5, dobbiamo ricorrere alla corrispondenza simbolica 27) che commenteremo a fianco delle singole istruzioni:

```
CLS ' pulisce lo schermo

INPUT " p " ; p ' richiesta di ingresso valore inferiore del campo di variabilità della x

INPUT " n " ; n  ' richiesta di ingresso valore superiore del campo di variabilità della x

INPUT " s " ; s  ' richiesta di ingresso valore dell'incremento di x

FOR x = p   TO n  STEP s   ' impone che la variabile indipendente (x) inizi il calcolo assumendo il
                           ' valore (p) , quando l'esecuzione del programma giunge alla istruzione
                           ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
                           ' valore (s) . Il ciclo si ripete con incrementi di (s) per arrestarsi
                           ' quando il valore di x ha raggiunto il valore (n)                        &
```

K = x ' uguaglianza di appoggio per l'istruzione successiva (non necessaria in questo caso)

Y = Y + K ' somma al valore di Y in memoria (uguale a zero all'inizio) il primo valore di K , e
' sostituisce in memoria il valore di Y=0 con il valore di Y=K , dopo la prima istruzione
' NEXT , x si incrementa di (s) e per conseguenza si incrementa anche K; il valore
' di Y in memoria si aggiorna sommandosi al nuovo valore di K, le somme si ripetono
' con ulteriori aggiornamenti del valore di Y in memoria fino a quando x raggiunge il
' valore dell'estremo superiore del campo di variabilità

NEXT x ' rimanda all'istruzione FOR per un successivo incremento di x

PRINT " Y = " ; Y ' comanda la presentazione del valore finale di Y

```
            F5
p ? 1
n ? 5
s ? .5
Y =  27
            F5
```

2.25 Esercizio di programmazione n° 17 (sommatoria algebrica progressiva)

Il risultato dell'esercizio precedente ci ha permesso di computare la sommatoria complessiva dei valori assunti da x nel campo di variabilità p = 1 n = 5 con incrementi di x pari a s =.5 definiti dalla seguente serie numerica 1; 1.5; 2; 2.5; 3; 3.5; 4; 4.5; 5. A volte è necessario conoscere anche i valori che la variabile dipendente Y assume mano a mano che la sommatoria avanza secondo gli incrementi assegnati alla x. Per far ciò è necessario modificare il programma del paragrafo 2.24 come segue (le modifiche sono indicate con il simbolo §):

CLS ' pulisce lo schermo

INPUT " p " ; p ' richiesta di ingresso valore inferiore del campo di variabilità della x

INPUT " n " ; n ' richiesta di ingresso valore superiore del campo di variabilità della x

INPUT " s " ; s ' richiesta di ingresso valore dell'incremento di x

FOR x = p TO n STEP s ' impone che la variabile indipendente (x) inizi il calcolo assumendo il
' valore (p) , quando l'esecuzione del programma giunge alla istruzione
' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
' valore (s) . Il ciclo si ripete con incrementi di (s) per arrestarsi
' quando il valore di x ha raggiunto il valore (n)

K = x ' uguaglianza di appoggio per l'istruzione successiva (non necessaria in questo caso)

Y = Y + K ' somma al valore di Y in memoria (uguale a zero all'inizio) il primo valore di K , e
' sostituisce in memoria il valore di Y=0 con il valore di Y=K , dopo la prima istruzione
' NEXT x si incrementa di (s) e di conseguenza si incrementa anche K; il valore
' di Y in memoria si aggiorna sommandosi al nuovo valore di K, le somme si ripetono
' con ulteriori aggiornamenti del valore di Y in memoria fino a quando x raggiunge il
' valore dell'estremo superiore del campo di variabilità

PRINT " x = " ; x , " Y = " ; Y ' § con questa istruzione si impone la presentazione dei valori di x
' dal primo all'ultimo affiancati ai valori progressivi assunti da Y,
' l'ultimo valore di Y rappresenta la sommatoria totale

NEXT x ' rimanda all'istruzione FOR per un successivo incremento di x

&

' § è stata omessa l'istruzione PRINT " Y = " ; Y

F5

```
p ? 1
n ? 5
s ? .5
x = 1      Y = 1
x = 1.5    Y = 2.5
x = 2      Y = 4.5
x = 2.5    Y = 7
x = 3      Y = 10
x = 3.5    Y = 13.5
x = 4      Y = 17.5
x = 4.5    Y = 22
x = 5      Y = 27
```

F5

2.26 Esercizio di programmazione n° 18 (sommatoria di funzione)

La sommatoria di funzione ha la stessa struttura della sommatoria algebrica, differisce da questa per la presenza di una funzione al posto della variabile indipendente; con la sommatoria di funzione si computa la somma di un certo numero di valori assunti dalla variabile dipendente quando la variabile indipendente varia nel campo assegnato ad incrementi stabiliti.
L'espressione che segue sintetizza il concetto:

$$Y = \sum_{x=p}^{x=n} f(x)$$

L'espressione mostra la sommatoria dei valori assunti da una generica funzione f (x) quando x assume i valori del proprio campo di variabilità ad iniziare da x = p per terminare con il valore x = n; nell'algoritmo non è specificata l'entità dell'incremento della variabile indipendente che viene invece definita nell'ambito della compilazione della routine di programma.
Iniziamo l'esercizio di programmazione scegliendo come funzione da utilizzare per la sommatoria l'esponenziale

$$K = e^{x}$$

nella quale la variabile dipendente è indicata con K per lasciare ad Y la rappresentanza della funzione sommatoria; vogliamo computare Y nel campo della variabile indipendente definito tra x = .1 ed x = 10 con incrementi dell'ordine di .2; con l'ausilio delle corrispondenze simboliche 28) e 16) compiliamo e commentiamo il nostro programma:

CLS ' pulisce lo schermo

INPUT " p " ; p ' richiesta di ingresso valore inferiore del campo di variabilità della x

INPUT " n " ; n ' richiesta di ingresso valore superiore del campo di variabilità della x

INPUT " s " ; s ' richiesta di ingresso valore dell'incremento di x

FOR x = p TO n STEP s ' impone che la variabile indipendente (x) inizi il calcolo assumendo il &

' valore (p) , quando l'esecuzione del programma giunge alla istruzione
' **NEXT,** il programma ritorna all'istruzione **FOR** che incrementa x del
' valore (s) . Il ciclo si ripete con incrementi di (s) per arrestarsi
' quando il valore di x ha raggiunto il valore (n)

K = EXP (x) ' funzione esponenziale soggetta alla sommatoria

Y = Y + K ' somma al valore di Y in memoria (uguale a zero all'inizio) il primo valore di K , e
' sostituisce in memoria il valore di Y=0 con il valore di Y=K , dopo la prima istruzione
' NEXT , x si incrementa di (s) e di conseguenza si incrementa anche K; il valore
' di Y in memoria si aggiorna sommandosi al nuovo valore di K. le somme si ripetono
' con ulteriori aggiornamenti del valore di Y in memoria fino a quando x raggiunge il
' valore dell'estremo superiore del campo di variabilità

NEXT x ' rimanda all'istruzione FOR per un successivo incremento di x

PRINT " Y = " ; Y ' comanda la presentazione del valore finale di Y

F5

```
p ? .1
n ? 10
s ? .2
Y =  109943.5
```

F5

L'esercizio ha portato, come era previsto, al computo della sommatoria complessiva di funzione; se si desiderano computare i valori progressivi della sommatoria si devono apportare al programma le semplici modifiche indicate in precedenza al paragrafo 2.25.

2.27 Come copiare un programma o parte di esso

Si rende necessario talvolta copiare un programma o parte di esso per ottenere un altro programma diverso dal primo per alcune varianti; ne è un esempio il caso citato alla fine del paragrafo precedente che propone la modifica del programma esistente. Se però si ha la necessità di tenere in memoria il programma originale è indispensabile farne una copia da modificare.
L'operazione di copiatura è semplice e consiste nelle seguenti operazioni:

- Si preme il tasto delle maiuscole e contemporaneamente con i tasti di direzione si evidenziano il programma o le righe di questo da copiare
- Si preme il tasto **CTRL** contemporaneamente al tasto **INS**
- Si passa ad una schermata " senza nome"
- Si preme il tasto delle maiuscole contemporaneamente al tasto **INS**

Dopo quest'ultima azione si ha sullo schermo la comparsa del programma o della parte di questo che è stata copiata; naturalmente il testo originale è rimasto integro.

2.28 Esercizio di programmazione n° 19 (la parabola)

Per questa esercitazione prendiamo la parabola come funzione rappresentativa delle operazioni matematiche che caratterizzano le funzioni della geometria analitica (iperbole esclusa); nella funzione parabola infatti sono presenti prodotti, elevamenti a potenza, somme e parametri .
La funzione parabola è rappresentata dall'espressione:

$$Y = a x^2 + bx + c$$

in cui le lettere a; b; c sono i parametri della funzione.

Procedendo al computo di questa funzione mediante la corrispondenza simbolica 30), assumendo il campo di variabilità della x compreso tra -5 e +5 con incrementi unitari, ed i seguenti parametri a = 1; b = 1; c = -2, con l'ausilio dell'istruzione di presentazione contemporanea di x e Y, si ha:

```
CLS ' pulizia dello schermo

FOR x= -5 TO 5 ' impone che la variabile indipendente (x) inizi il calcolo assumendo il
               ' valore (-5) , quando l'esecuzione del programma giunge alla istruzione
               ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
               ' valore (1) . Il ciclo si ripete con incrementi di (1) per arrestarsi
               ' quando il valore di x ha raggiunto il valore (5)

Y = x ^ 2 + x - 2 ' funzione parabola da computare

PRINT " x = " ; x , "Y = " ; Y ' comando visualizzazione dati variabile indipendente associati ai
                               ' corrispondenti valori variabile dipendente

NEXT x ' rimanda all'istruzione FOR per un incremento successivo della x
```

F5

```
x = -5     Y = 18
x = -4     Y = 10
x = -3     Y = 4
x = -2     Y = 0
x = -1     Y = - 2
x = 0      Y = - 2
x = 1      Y = 0
x = 2      Y = 4
x = 3      Y = 10
x = 4      Y = 18
x = 5      Y = 28
```

F5

I risultati di questo esercizio mostrano una caratteristica particolare della funzione parabola, caratteristica per cui, in dipendenza dei valori dei parametri, la variabile dipendente si azzera due volte nel campo di variabilità della x; nel nostro caso gli zeri di Y si ottengono per i valori x = - 2 ed x = 1.

Come è noto questi due valori di x sono le soluzioni dell'equazione di 2° grado che si ottiene dalla funzione parabola uguagliandola a zero.

2.29 Esercizio di programmazione n° 20 (l' iperbole)

La computazione della funzione iperbole richiede attenzione particolare dato che essa è infinitamente grande per x = 0. E' chiaro che se noi tentiamo il computo della funzione per x = 0 il Qbasic blocca l'esecuzione del calcolo indicando " **Divisione per zero**". E' pertanto indispensabile che il campo di variabilità della x non comprenda valori tanto piccoli tali da portare il computer a denunciare tale condizione.

Ciò premesso possiamo compilare il programma in base alla corrispondenza simbolica 32) ed al sistema automatico a campo fisso per m = .1 n = 1.91 s = .2 .

```
CLS ' pulizia dello schermo

FOR x = .1 TO 1.91 STEP .2 ' impone che la variabile indipendente (x) inizi il calcolo assumendo      &
```

```
                    ' il  valore (.1) , quando l'esecuzione del programma giunge alla istruzione
                    ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
                    ' valore (.2) . Il ciclo si ripete con incrementi di (.2) per arrestarsi
                    ' quando il valore di x ha raggiunto il valore (1.9)

Y =  1 / x   ' funzione iperbole da computare

PRINT " Y = " ; Y  ' comando visualizzazione dati variabile dipendente

NEXT x  ' rimanda all'istruzione FOR per un incremento successivo della x


              F5
Y=10
Y=3.333333
Y=2
Y=1.428571
Y=1.111111
Y=.9090909
Y=.7692307
Y=.6666666
Y=.5882353
Y=.5263157
              F5
```

2.30 Esercizio di programmazione n° 21 (operatori aritmetici relazionali)

Gli operatori aritmetici relazionali 8) e 9) introducono importanti ampliamenti delle capacità operative del sistema di calcolo in generale; mostriamo di seguito un piccolo esempio di ciò che si può fare con essi, lasciando al lavoro seguente ed alle capacità inventive del lettore che avrà maturato esperienza, le innumerevoli applicazioni possibili.
Riprendiamo allo scopo l'esercizio n° 1 relativo al calcolo della funzione:

$$Y = (x)^{1/2}$$

sviluppato per tre valori della variabile indipendente x = 3.75; x = 7; x = 12, un campo di variazione compreso tra i valori positivi 3.75 e 12; se il campo di variabilità della x fosse stato compreso tra - 3.75 e + 12 l'esercizio non sarebbe stato possibile, nell'ambito dei numeri reali, per l'estremo inferiore del campo, essendo esso un numero negativo. Ora con l'ausilio degli operatori aritmetici relazionali è possibile risolvere **"formalmente"** la funzione per qualsiasi campo di variabilità della x; ottenendo valori reali di Y per valori positivi di x e valori immaginari di Y per valori negativi di x.
Per giungere al risultato voluto è anzitutto necessario trasformare la funzione in oggetto mediante l'imposizione del valore assoluto 7) sotto radice:

$$Y = (| x |)^{1/2}$$

che consente il calcolo della funzione indipendentemente dal segno di x; ora mediante le corrispondenze simboliche 1) e 7) si può scrivere la funzione radice in valore assoluto della x come segue:

Y = SQR (ABS (x))

a questo punto non resta che compilare il nuovo programma che utilizza gli operatori relazionali e le istruzioni ad essi associate, istruzioni che saranno opportunamente commentate nel programma

stesso, per il computo della funzione nel campo di variabilità della x compreso tra -5 e + 5; i risultati attesi sono numeri reali e numeri immaginari, quest'ultimi distinti dai primi dal simbolo **j**:

```
CLS ' pulizia dello schermo

FOR x = -5 TO +5 ' impone che la variabile indipendente (x) inizi il calcolo assumendo il
                 ' valore ( -5 ) ,quando l'esecuzione del programma giunge alla istruzione
                 ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
                 ' valore ( 1 ) . Il ciclo si ripete con incrementi di ( 1 ) per arrestarsi
                 ' quando il valore di x ha raggiunto il valore ( + 5 )

Y = SQR ( ABS ( x ) ) ' funzione radice con x in valore assoluto

IF x < 0 THEN PRINT " j " ; Y ' l'istruzione impone che se x è inferiore a zero venga stampato ,
                              ' prima del valore numerico della variabile dipendente ,il simbolo
                              ' j per indicare che è stata sviluppata (simbolicamente) una radice
                              ' quadrata di un numero negativo e che il risultato è un numero
                              ' immaginario puro

IF x >= 0 THEN PRINT ; Y ' l'istruzione impone che se x è maggiore od uguale a zero il valore
                         ' numerico della variabile dipendente venga stampato in modo
                         ' normale dato che esso è il risultato della radice quadrata di un
                         ' numero positivo

NEXT x ' comanda il ritorno all'istruzione FOR per gli incrementi successivi.
```

F5

```
j 2.236068
j 2
j 1.732051
j 1.414214
j 1
0
1
1.414214
1.732051
2
2.236068
```

F5

I risultati mostrano che la funzione in esame ha fornito cinque valori immaginari e sei valori reali di Y come la teoria prevede.
L'esercizio che abbiamo condotto si basa, sia sugli operatori aritmetici relazionali, sia sulle istruzioni **IF** x > a **THEN** ..... che sono estesamente trattate nei manuali del Qbasic; non rientrando negli obiettivi di questo testo un approfondimento su tale argomento si rimanda il lettore ai riferimenti bibliografici riportati nelle ultime pagine.

2.31 Sulle funzioni a due valori

Negli esercizi che abbiamo svolto non sono state elaborate numericamente le funzioni a due valori 31); 33); 34) relative alla geometria analitica; dato che per esse è significativa soltanto la presentazione grafica, ci riserviamo di trattarle in tal modo nel successivo capitolo 3.

CAPITOLO 3

LA GRAFICA E GLI ESERCIZI PER LA VISUALIZZAZIONE DELLE FUNZIONI MATEMATICHE

Questo capitolo è dedicato alla tecnica di programmazione necessaria per la visualizzazione dei grafici relativi alle funzioni matematiche, funzioni che possono essere fini a se stesse per gli sviluppi puramente analitici, e che possono inoltre rappresentare utilmente gli andamenti di innumerevoli fenomeni fisici. Le spiegazioni sulla programmazione sono accompagnate da esercizi da svolgere direttamente sul P.C. In questo capitolo non si richiamano più le corrispondenze simboliche tra le notazioni ordinarie e le notazioni Qbasic che lasciamo come esercizio al lettore.

3.1 L'obiettivo della grafica

Il tracciamento dei grafici delle funzioni favorisce lo studio e l'analisi minuziosa delle leggi che rappresentano; è un indispensabile ausilio per la migliore comprensione dei problemi, sia di carattere puramente didattico, sia di carattere tecnico.
Il metodo permette di superare facilmente e rapidamente le naturali difficoltà insite nello studio e nell'applicazione della matematica, esso rende tangibile ciò che a volte può apparire astratto.
La grafica che ci accingiamo a trattare ha un obiettivo preciso: la visualizzazione a colori delle curve rappresentative dell'andamento delle funzioni matematiche.
Come è noto le funzioni sono esprimibili graficamente, sia in coordinate polari, sia in coordinate cartesiane, noi ci interesseremo della presentazione di tipo cartesiano che è usata più frequentemente.
Per le ragioni sopra esposte è chiaro che dobbiamo anzitutto imparare come costruire il sistema di assi cartesiani nell'ambito del quale, successivamente, tracciare le curve caratteristiche delle funzioni che ci interessano; pertanto il primo passo da fare è relativo alla conoscenza delle capacità grafiche del Qbasic che rendono la presentazione delle funzioni sullo schermo video del P.C.

3.2 Le capacità grafiche del Qbasic e lo schermo video

Le capacità grafiche del Qbasic sono governate da numerose istruzioni di programma la cui descrizione completa ci porterebbe lontano dal nostro obiettivo, rimandando il lettore interessato ad ampliare questo argomento ai testi citati nella bibliografia, ci limitiamo ad esporre soltanto quanto è necessario ed indispensabile ai nostri scopi.
La prima informazione da dare al lettore è relativa all'istruzione specifica che deve precedere qualsiasi programma di grafica indirizzato al nostro lavoro; da questa istruzione dipende tanto il formato della presentazione quanto il numero di punti elementari **(pixel)** che sono contenibili nel formato stesso. L'istruzione in oggetto, per la modalità grafica ad alta risoluzione, è:

SCREEN 9

se il programma per la composizione grafica inizia con questa istruzione è fissato il formato a tutto schermo della presentazione ed il seguente numero di punti è contenibile all'interno del formato:

224000 pixel

questa capacità grafica è determinata dai due valori di definizione orizzontale e verticale pari a:

640 pixel nel senso orizzontale dello schermo
350 pixel nel senso verticale dello schermo

Inoltre l'eventuale scrittura di caratteri nello schermo è definita con un numero di caratteri pari a 80 in orizzontale e 24 in verticale, ciascuno dalle dimensioni di 8 x 14 pixel.

3.3 La presentazione dei punti sullo schermo video

Una volta fissata in **SCREEN 9** la prima istruzione del programma dobbiamo eseguire alcuni esercizi per imparare come si presentano gli elementi fondamentali della grafica: **i punti** .
Ciascun punto è definito mediante le proprie coordinate cartesiane assolute che fanno riferimento alla posizione del "**punto zero**" posto nell'angolo sinistro della parte superiore dello schermo.
L'istruzione che individua sullo schermo un qualsiasi punto contiene le sue coordinate (x ; y) rispetto al **punto zero**, l'istruzione generica per la presentazione di un punto è la seguente:

PSET (x , y)

ne consegue che il **punto zero** ha come istruzione che lo visualizza

PSET (0 , 0)

Dato che il sistema grafico prevede che i valori delle coordinate dei punti debbano essere espresse in pixel, precisiamo:
-alla coordinata x si possono attribuire valori compresi tra 0 e 639 (infatti il numero massimo dei pixel nella dimensione orizzontale dello schermo è 640)
-alla coordinata y si possono attribuire valori compresi tra 0 e 349 (infatti il numero massimo dei pixel nella dimensione verticale dello schermo è 350)
E' necessario inoltre specificare che le coordinate dei punti in Qbasic, diversamente dai punti della geometria analitica, devono sempre essere espresse mediante numeri positivi, ciò per convenzione del sistema grafico.

3.4 Esercitazione grafica n° 1 (presentazione di punti)

Con le due istruzioni di cui siamo in possesso compiliamo un breve programma grafico che ci dia la possibilità di presentare cinque punti luminosi sullo schermo. P1 (punto zero) di coordinate (0;0), P2 (punto all'estremo destro alto) di coordinate (639;0), P3 (punto tutto sinistra in basso) di coordinate (0;190), P4 (punto tutto destra in basso) di coordinate (639;190), P5 (punto interno) di coordinate (100;80); partendo da una schermata vuota compiliamo:

```
SCREEN 9        ' impostazione modalità grafica di schermata
PSET(0,0)       ' presentazione di P1
PSET(639,0)     ' presentazione di P2
PSET(0,190)     ' presentazione di P3
PSET(639,190)   ' presentazione di P4
PSET(100,80)    ' presentazione di P5
                F5
```

si ha la presentazione dei cinque punti voluti a luce bianca su sfondo nero, compare inoltre in fondo allo schermo la scritta " **premere un tasto per continuare**".

F5

Per evitare che i grafici vadano ad interferire con la scritta menzionata non utilizzeremo mai tutti i 350 pixel di definizione verticale ma ci limiteremo a 320 pixel.

3.5 Esercitazione grafica n° 2 (colorazione dei punti)

Il Qbasic in modalità grafica SCREEN 9 permette la presentazione dei punti in 15 colori diversi, ciascuno distinto da un numero di codice che deve essere posizionato a fianco dell'istruzione PSET per assegnarle il colore voluto:

1-BLU
2-VERDE
3-TURCHESE
4-ROSSO
5-MAGENTA
6-MARRON
7-BIANCO
8-GRIGIO
9-AZZURRO
10-VERDE
11-TURCHESE CHIARO
12-ROSSO CHIARO
13-MAGENTA CHIARO
14-GIALLO
15-BIANCO LUMINOSO

L'istruzione PSET si completa con il codice colore come nei due esempi sotto riportati:
-per ottenere il verde si aggiunge all'istruzione il numero 2: PSET(x,y),2
-per ottenere il rosso si aggiunge all'istruzione il numero 4: PSET(x,y),4
se l'istruzione non porta nessun codice di colore il punto risulta bianco luminoso come se fosse riportato il codice 15.
E' immediato perciò compilare un secondo programma che esegua la presentazione di punti luminosi colorati:

```
SCREEN 9              ' impostazione modalità di schermata
PSET(0,0),2           ' presentazione P1 di colore verde
PSET(639,0)           ' presentazione P2 di colore bianco luminoso
PSET(0,190),7         ' presentazione P3 di colore bianco
PSET(639,190),14      ' presentazione P4 di colore giallo
PSET(100,80),11       ' presentazione P5 di colore turchese chiaro
```

F5

si ha la presentazione dei cinque punti voluti su sfondo nero, compare inoltre in fondo allo schermo la scritta bianca" **premere un tasto per continuare**".

F5

3.6 La presentazione delle rette sullo schermo

Per la costruzione grafica degli assi cartesiani è d'obbligo tracciare due rette ortogonali che dividono il piano in quattro quadranti, poi si completa il disegno del sistema con le tracce relative

alle suddivisioni degli assi X e Y, suddivisioni che possono prolungarsi nei quattro quadranti per formare un reticolo simile a quello stampato sulla carta millimetrata.
Elemento fondamentale per questo tipo di costruzione è la presentazione delle rette sullo schermo; ciascuna retta è definita mediante le coordinate cartesiane assolute di due punti che fanno riferimento alla posizione del "**punto zero**".
L'istruzione che individua una qualsiasi retta nello schermo contiene le coordinate (x1 ; y1) e (x2 ; y2) degli estremi rispetto al **punto zero**, l'istruzione generica per la presentazione di una retta è la seguente:

LINE (x1, y1) - (x2,y2)

ne segue che la retta che ha come estremi il punto zero e l'angolo superiore destro dello schermo è visualizzata dall'istruzione:

LINE (0 , 0) - (639 , 0)

Se ora facciamo riferimento all'esercitazione grafica n° 1 possiamo dire che questa retta è la congiungente i due punti P1 e P2.
Similmente le rette definite dalle istruzioni

LINE (0 , 0) - (0 , 190)
LINE (639 , 0) - (639 , 190)
LINE (0 , 190) - (639 , 190)

sono rispettivamente le congiungenti dei punti P1 P3
P2 P4
P3 P4

Le rette prese in considerazione rappresentano pertanto i quattro lati di un rettangolo che si può presentare sullo schermo sì da escludere dalla sua area la scritta "premere un tasto per continuare".
La grafica Qbasic consente di tracciare le rette a colori nello stesso modo e con gli stessi codici impiegati per la presentazione dei punti; è sufficiente corredare l'istruzione nel seguente modo:

LINE (x1 , y1) - (x2 , y2)	traccia bianca luminosa
LINE (x1 , y1) - (x2 , y2),2	traccia verde
LINE (x1 , y1) - (x2 , y2),14	traccia gialla

3.7 Esercitazione grafica n° 3 (presentazione di rette e punti)

Utilizzando le istruzioni riportate nei paragrafi 3.5 e 3.6 compiliamo il programma grafico che ci dà modo di costruire un rettangolo all'interno del quale compaiono tre punti collocati a caso:

```
SCREEN 9                        ' impostazione modalità grafica
LINE (0,0)-(639,0)              ' traccia la retta tra P1 e P2, colore bianco luminoso
LINE (0,0)-(0,190),1            ' traccia la retta tra P1 e P3, colore blu
LINE (639,0)-(639,190),2        ' traccia la retta tra P2 e P4, colore verde
LINE (0,190)-(639,190),5        ' traccia la retta tra P3 e P4, colore magenta
PSET(50,80),3          ' presentazione punto di colore turchese                &
```

PSET(100,130),9 ' presentazione punto di colore azzurro

PSET(160, 100),14 ' presentazione punto di colore giallo

F5

si ha la presentazione di un rettangolo con i lati diversamente colorati, all'interno del quale compaiono tre punti colorati.

F5

3.7.1 Specificazioni sui punti e sulle rette

Una fondamentale osservazione deve essere fatta in merito al tracciamento dei punti e delle rette: avendo assunto per il nostro lavoro la modalità grafica SCREEN 9 le coordinate x ed y dei punti singoli e degli estremi delle rette, riferiti al punto zero, portano a spostamenti geometrici sullo schermo diversi a parità di numero di pixel assegnati ad x ed a y.
Ciò è facilmente verificabile sul P.C. seguendo il semplice programma di controllo che pone tre punti sullo schermo:

SCREEN 9

PSET (0, 0) ' traccia il punto Po (punto zero) nell'angolo superiore sinistro dello schermo

PSET (0, 100) ' traccia il punto P1 a 100 pixel sotto il punto zero

PSET (100, 0) ' traccia il punto P2 a 100 pixel a fianco del punto zero

F5

La presentazione dei tre punti mostra chiaramente cha la distanza tra Po e P1 è molto più grande della distanza tra Po e P2. Il rapporto tra le due distanze è circa 1.44; ciò significa che se si desidera tracciare P2 alla stessa distanza di P1 rispetto a Po l'ascissa di P2 deve essere moltiplicata per il suddetto rapporto: x di P2 $= 1.44 \cdot 100 = 144$.
Se ripetiamo il programma di controllo con questa nuova ascissa di P2 vediamo che i punti P1 e P2 sono ora equidistanti da Po:

SCREEN 9

PSET (0, 0) ' traccia il punto Po (punto zero) nell'angolo superiore sinistro dello schermo

PSET (0, 100) ' traccia il punto P1 a 100 pixel sotto il punto zero

PSET (144, 0) ' traccia il punto P2 a 144 pixel a fianco del punto zero

F5

Di questo importante aspetto della grafica deve essere tenuto conto ogni qualvolta si renda necessario costruire tracciati ed altro; in particolare nella fase di costruzione del sistema di assi cartesiani che deve avere assi della stessa lunghezza e deve essere suddiviso da un reticolo a maglie quadrate.

3.8 Le potenzialità dell'istruzione PSET(x , y)

L'istruzione PSET(x , y), impiegata in precedenza semplicemente per collocare punti luminosi sullo schermo, possiede delle potenzialità che sono il fondamento di tutta la grafica Qbasic indirizzata alla presentazione delle funzioni matematiche in generale. L'istruzione in oggetto è

infatti una "funzione grafica"; essa governa la posizione dei punti sullo schermo al variare delle coordinate x; y, variazione che può avvenire nel corso di esecuzione di un programma permettendo il posizionamento successivo di una serie di punti, che seguono pertanto traiettorie prestabilite.
Grazie al sistema di memoria del P.C. tutti i punti che hanno seguito una traiettoria restano visibili in permanenza realizzando la composizione sullo schermo di qualsiasi tracciato.
Per variare nel modo più semplice una delle due coordinate di un punto durante l'esecuzione di un programma si può utilizzare il "Sistema automatico per il calcolo di funzioni a campo fisso" di cui al paragrafo 2.11.
Per variare contemporaneamente le due coordinate di un punto durante l'esecuzione di un programma si può impiegare il "Sistema automatico per il calcolo delle funzioni di più variabili in campi diversi" illustrato nel paragrafo 2.19.
Nelle esercitazioni a seguire compileremo alcuni programmi che, avvalendosi dei due sistemi automatici citati, ci permetteranno di visualizzare sullo schermo particolari traiettorie di punti.

3.9 Esercitazione grafica n° 4 (punteggiata verticale)

Sulla base di quanto scritto nel paragrafo 3.8 iniziamo ad utilizzare le importanti proprietà dell'istruzione PSET(x , y) sviluppando la seguente esercitazione:
vogliamo tracciare una punteggiata verticale che parte dal punto P1(0;0) e termina al punto P2(0;160); se procedessimo per singoli punti, così come mostrato al paragrafo 3.4, dovremmo digitare 80 istruzioni PSET(x , y) tutte con valore x = 0 e valori di y crescenti 2; 4; 6 ..; 160. Si risolve invece il nostro problema, semplicemente, compilando il seguente programma con l'ausilio del Sistema automatico di calcolo di paragrafo 2.11:

```
SCREEN 9

FOR y = 0 TO 160 STEP 2 ' impone che la coordinata (y) inizi il calcolo assumendo il
                        ' valore (0) , quando l'esecuzione del programma giunge alla istruzione
                        ' NEXT il programma ritorna all'istruzione FOR che incrementa y del
                        ' valore (2) . Il ciclo si ripete con incrementi di (2) per arrestarsi
                        ' quando il valore di y ha raggiunto il valore (160)

PSET (0 , y) ' comanda la presentazione di tutti i punti che hanno ascissa x = 0 e ordinate
             ' variabili secondo l'istruzione FOR

NEXT y ' comanda il programma al ritorno automatico alla istruzione FOR
```

F5

si ha la presentazione sulla parte sinistra dello schermo di una punteggiata verticale bianca luminosa che si estende tra il punto zero ed il centosessantesimo pixel.

F5

Dopo il ritorno alla schermata di programma si può aggiungere il colore all'istruzione PSET e modificare a piacere gli estremi del campo di variabilità di y per fare altra esperienza sul metodo.
Si deve osservare che il passo di incremento della y, fissato nell'esercizio in 2 pixel, può essere aumentato come si desidera; non può invece essere diminuito (ad 1) se non si vuole che la punteggiata si trasformi in una linea continua. Infatti, dato che ad ogni unità di incremento corrisponde un pixel, se l'incremento è di 2 pixel si ha un punto luminoso intervallato da un pixel non illuminato, diversamente si hanno tutti i pixel contigui illuminati.

3.10 Esercitazione grafica n° 5 (punteggiata orizzontale)

Proponiamoci di tracciare una punteggiata orizzontale che parte dal punto P1 (0;0) e termina al punto P2 (200;0). Si risolve il nostro problema compilando il programma mediante l'imposizione della variabilità di x:

```
SCREEN 9

FOR  x = 0  TO 200 STEP 2  ' impone che la coordinata (x) inizi il calcolo assumendo il
                           ' valore (0) , quando l'esecuzione del programma giunge alla istruzione
                           ' NEXT il programma ritorna all'istruzione FOR che incrementa x del
                           ' valore (2) . Il ciclo si ripete con incrementi di (2) per arrestarsi
                           ' quando il valore di x ha raggiunto il valore (200)

PSET (x , 0) ,1  ' comanda la presentazione di tutti i punti che hanno ordinata y = 0 e ascisse
                 ' variabili secondo l'istruzione FOR (colore blu)

NEXT x  ' comanda il programma al ritorno automatico alla istruzione FOR
```

F5

si ha la presentazione sulla parte alta dello schermo di una punteggiata orizzontale blu che si estende tra il punto zero ed il duecentesimo pixel.

F5

Si suggerisce, una volta tornati alla schermata di programma, di modificare i valori numerici dati per prendere maggiore confidenza con il metodo.

3.11 Esercitazione grafica n° 6 (punteggiate verticali parallele)

Il problema grafico che ora ci accingiamo a risolvere presenta un grado di difficoltà superiore ai casi precedenti ma è di fondamentale importanza per la costruzione dei sistemi di assi cartesiani necessari al nostro lavoro. Dobbiamo tracciare, contemporaneamente, 6 punteggiate verticali parallele, equidistanti 20 pixel, partendo dalla punteggiata collocata tra P1 (0;0) e P2 (0;100); compiliamo il programma con l'ausilio del "Sistema automatico di calcolo delle funzioni di più variabili " illustrato nel paragrafo 2.19:

```
SCREEN 9

FOR  x = 0  TO  100  STEP  20  'impone che l'ascissa (x) inizi il calcolo assumendo il valore (0)
                               ' resta poi in attesa che l'istruzione successiva costruisca la prima
                               ' punteggiata. Dopo che la prima punteggiata è stata tracciata
                               ' il programma giunge alla istruzione NEXT x che riporta la
                               ' esecuzione su FOR x = .. che incrementa x di 20 pixel,
                               ' il processo si ripete fino al valore di x = 100 per il tracciamento
                               ' dell'ultima punteggiata.

FOR  y = 0  TO  100  STEP 2  ' impone che l'ordinata (y) inizi il calcolo assumendo il valore (0)
                             ' quando l'esecuzione del programma giunge alla istruzione
                             ' NEXT y il programma ritorna all'istruzione FOR y = che incrementa y
                             ' di 2 pixel . Il ciclo si ripete con incrementi di (2) per arrestarsi
                             ' quando il valore di y ha raggiunto il valore (100) con
                             ' l' ultimazione della prima punteggiata
                             ' Il programma passa quindi all'istruzione NEXT x che lo invia a FORx=
                             ' per l'impostazione della punteggiata successiva                &
```

PSET (**x, y**) ' comanda il posizionamento dei punti in base alle due istruzioni **FOR**

NEXT y ' comanda il programma al ritorno automatico alla istruzione **FOR** y =

NEXT x ' comanda il programma al ritorno automatico alla istruzione **FOR** x =

F5

si ha la presentazione di 6 punteggiate verticali bianche distanti l'una dall'altra di 20 pixel

F5

Data la particolare caratteristica di questa esercitazione si consiglia, una volta tornati alla schermata di programma, di modificare i valori numerici dell'esercizio in base a nuove configurazioni geometriche da implementare dopo averle elaborate a tavolino con carta e matita.

3.12 Esercitazione grafica n° 7 (punteggiate orizzontali parallele)

Utilizzando lo stesso metodo dell'esercitazione precedente compiliamo un programma in grado di tracciare, contemporaneamente, 6 punteggiate orizzontali parallele, equidistanti 20 pixel, partendo dalla punteggiata collocata tra P1 (0;0) e P3 (100;0):

```
SCREEN 9

FOR y = 0 TO 100 STEP 20 ' impone che l'ordinata (y) inizi il calcolo assumendo il valore (0)
                         ' resta poi in attesa che l'istruzione successiva costruisca la prima
                         ' punteggiata. Dopo che la prima punteggiata è stata tracciata
                         ' il programma giunge alla istruzione NEXT y che riporta la
                         ' esecuzione su FOR y = che incrementa y di 20 pixel,
                         ' il processo si ripete fino al valore di y = 100 per il tracciamento
                         ' dell'ultima punteggiata.

FOR x = 0 TO 100 STEP 2 ' impone che l'ascissa (x) inizi il calcolo assumendo il valore (0)
                        ' quando l'esecuzione del programma giunge alla istruzione
                        ' NEXT x il programma ritorna all'istruzione FOR x= che incrementa x
                        ' di 2 pixel . Il ciclo si ripete con incrementi di (2) per arrestarsi
                        ' quando il valore di x ha raggiunto il valore (100) con la
                        ' ultimazione della prima punteggiata
                        ' Il programma passa quindi all'istruzione NEXT y che lo invia a FORy=
                        ' per l'impostazione della punteggiata successiva

PSET ( x, y ) ' comanda il posizionamento dei punti in base alle due istruzioni FOR

NEXT x ' comanda il programma al ritorno automatico alla istruzione FOR x =

NEXT y ' comanda il programma al ritorno automatico alla istruzione FOR y=
```

F5

si ha la presentazione di 6 punteggiate orizzontali bianche distanti l'una dall'altra di 20 pixel

F5

3.13 Esercitazione grafica n° 8 (reticolo di punteggiate)

Utilizzando i due programmi sviluppati nei paragrafi 3.11 e 3.12 è facile compilare un nuovo programma che dà modo di tracciare sullo schermo un reticolo di punteggiate.

Il reticolo ha maglie rettangolari con i lati maggiori disposti in verticale, ciò per la ragione evidenziata nel paragrafo 3.7.1.
Il nuovo programma conduce al tracciamento di un reticolo che ha due lati contigui rispettivamente tra P1(0;0) e P2(0;100); tra P1(0;0) e P3(100;0).

```
SCREEN 9

FOR x = 0 TO 100 STEP 20 ' impone che l'ascissa (x) inizi il calcolo assumendo il valore (0)
                         ' resta poi in attesa che l'istruzione successiva costruisca la prima
                         ' punteggiata verticale. Dopo che la prima punteggiata è stata tracciata
                         ' il programma giunge alla prima istruzione NEXT x che riporta la
                         ' esecuzione su il primo FOR x = che incrementa x di 20 pixel,
                         ' il processo si ripete fino al valore di x = 100 per il tracciamento
                         ' dell'ultima punteggiata verticale.
FOR y = 0 TO 100 STEP 2  ' impone che l'ordinata (y) inizi il calcolo assumendo il valore (0)
                         ' quando l'esecuzione del programma giunge alla prima istruzione
                         ' NEXT y, il programma ritorna alla prima istruzione FOR y= che incrementa y
                         ' di 2 pixel . Il ciclo si ripete con incrementi di (2) per arrestarsi
                         ' quando il valore di y ha raggiunto il valore (100) con la
                         ' ultimazione della prima punteggiata verticale
                         ' Il programma passa quindi alla prima istruzione NEXT x che lo invia
                         ' alla prima istruzione FORx= per l'impostazione della punteggiata successiva
PSET ( x, y ) ' comanda il posizionamento dei punti per il tracciamento delle punteggiate verticali
              ' in base alla prima coppia di istruzioni FOR

NEXT y ' comanda il programma al ritorno automatico alla prima istruzione FOR y =

NEXT x ' comanda il programma al ritorno automatico alla prima istruzione FOR x =

FOR y = 0 TO 100 STEP 20 ' impone che l'ordinata (y) inizi il calcolo assumendo il valore (0)
                         ' resta poi in attesa che l'istruzione successiva costruisca la prima
                         ' punteggiata orizzontale. Dopo che la prima punteggiata è stata tracciata
                         ' il programma giunge alla seconda istruzione NEXT y che riporta
                         ' l'esecuzione sul secondo FOR y = che incrementa y di 20 pixel,
                         ' il processo si ripete fino al valore di y = 100 per il tracciamento
                         ' dell'ultima punteggiata orizzontale.

FOR x = 0 TO 100 STEP 2  ' impone che l'ascissa (x) inizi il calcolo assumendo il valore (0)
                         ' quando l'esecuzione del programma giunge alla seconda istruzione
                         ' NEXT x , il programma ritorna alla seconda istruzione FOR x= che
                         ' incrementa x di 2 pixel . Il ciclo si ripete con incrementi di (2) per arrestarsi
                         ' quando il valore di x ha raggiunto il valore (100) con la
                         ' ultimazione della prima punteggiata orizzontale
                         ' Il programma passa quindi alla seconda istruzione NEXT y che lo invia
                         ' al secondo FORy= per l'impostazione della punteggiata successiva

PSET ( x, y ) ' comanda il posizionamento dei punti per il tracciamento delle punteggiate orizzontali
              ' in base alla seconda coppia delle istruzioni FOR

NEXT x ' comanda il programma al ritorno automatico alla seconda istruzione FOR x =

NEXT y ' comanda il programma al ritorno automatico alla seconda istruzione FOR y=
```

F5

si ha la presentazione del reticolo di punteggiate voluto nella zona dello schermo in alto a sinistra

3.14 La formazione del sistema di assi cartesiani

La formazione del sistema di assi cartesiani, necessario alla rappresentazione grafica delle funzioni matematiche, si può avvalere del programma compilato al paragrafo 3.13; tale sistema infatti consiste in un reticolo, simile a quello mostrato nell'esercitazione n° 8, opportunamente dimensionato e suddiviso, in cui compaiono gli assi ortogonali x ed y .
Per semplificare il tracciamento delle funzioni nel reticolo si fissano per questo, ricordando quanto specificato nel paragrafo 3.7.1, le seguenti caratteristiche geometriche:

- POSIZIONAMENTO DEL RETICOLO CON LO SPIGOLO SINISTRO COINCIDENTE CON IL PUNTO Po (0 ; 0) (punto zero)

- RAPPORTO PIXEL PER OTTENERE IL RETICOLO A MAGLIE QUADRATE X / Y = 23 / 16

- DIMENSIONE ORIZZONTALE PARI A 460 PIXEL

- DIMENSIONE VERTICALE PARI A 320 PIXEL

- DIVISIONE ORIZZONTALE A 20 INTERVALLI DI 23 PIXEL CIASCUNO

- DIVISIONE VERTICALE A 20 INTERVALLI DI 16 PIXEL CIASCUNO

- ASSE DELLE ASCISSE DI LUNGHEZZA PARI A 460 PIXEL POSIZIONABILE IN DUE MODI DIVERSI:
NELLA MEZZERIA VERTICALE DEL RETICOLO (per tracciamento funzioni ordinarie)
SULL' ESTREMO INFERIORE DEL RETICOLO (per tracciamento funzioni con y positivo)

- ASSE DELLE ORDINATE DI LUNGHEZZA PARI A 320 PIXEL POSIZIONABILE IN DUE MODI DIVERSI:
NELLA MEZZERIA ORIZZONTALE DEL RETICOLO (per tracciamento funzioni ordinarie)
SULL' ESTREMO SINISTRO DEL RETICOLO (per tracciamento funzioni pari)

-COORDINATE DELL' ORIGINE "O" DEGLI ASSI RISPETTO AL PUNTO Po: O (230;160)

- COLORE DELLE TRACCE RETICOLO = BIANCO

- COLORE TRACCE ASSI ORTOGONALI = BIANCO LUMINOSO

Il programma completo per la presentazione del sistema di assi cartesiani è qui di seguito compilato e commentato; esso deve essere provato sul P.C. e successivamente memorizzato, ad esempio con il nome CARTES.BAS, per poterlo richiamare ogni qualvolta un programma matematico richieda la rappresentazione grafica dei risultati numerici.

```
' PROGRAMMA PER LA COSTRUZIONE DI UN SISTEMA DI ASSI CARTESIANI

SCREEN 9

FOR x = 0 TO 460 STEP 23 ' impone che l'ascissa (x) inizi il calcolo assumendo il valore (0)
                         ' resta poi in attesa che l'istruzione successiva costruisca la prima
                         ' punteggiata. Dopo che la prima punteggiata è stata tracciata
                         ' il programma giunge la prima istruzione NEXT x che riporta la
                         ' esecuzione su il primo FOR x = che incrementa x di 23 pixel,
                         ' il processo si ripete fino al valore di x = 460 per il tracciamento
                         ' dell'ultima punteggiata verticale.

FOR y = 0 TO 320 STEP 2 ' impone che l'ordinata (y) inizi il calcolo assumendo il valore (0)          &
```

```
                        ' quando l'esecuzione del programma giunge alla prima istruzione
                        ' NEXT y, il programma ritorna alla prima istruzione FOR y= che incrementa y
                        ' di 2 pixel . Il ciclo si ripete con incrementi di (2) per arrestarsi
                        ' quando il valore di y ha raggiunto il valore (320)  con la
                        ' ultimazione della  prima punteggiata verticale
                        ' Il programma passa  quindi alla prima istruzione NEXT x che lo invia
                        ' al primo FORx= per l'impostazione della punteggiata successiva

PSET ( x, y ),7  ' comanda il posizionamento dei punti per il tracciamento delle punteggiate verticali
                 ' in base alla prima coppia di istruzioni FOR (colore della punteggiata = bianco)

NEXT  y  ' comanda il programma al ritorno automatico alla prima istruzione FOR  y =

NEXT  x  ' comanda il programma al ritorno automatico alla  prima istruzione  FOR x =

FOR  y = 0  TO  320  STEP  16  ' impone che  l'ordinata  (y) inizi il calcolo assumendo il valore  (0)
                               ' resta poi in attesa che l'istruzione successiva costruisca la prima
                               ' punteggiata. Dopo che la prima punteggiata è stata tracciata
                               ' il programma giunge alla seconda istruzione NEXT y  che riporta
                               ' l'esecuzione sul secondo FOR y = che incrementa  y di 16 pixel,
                               ' il processo si ripete fino al valore di y = 320  per il tracciamento
                               ' dell'ultima punteggiata orizzontale.

FOR  x = 0  TO  460  STEP 3  ' impone che l'ascissa (x) inizi il calcolo assumendo il valore (0)
                             ' quando l'esecuzione del programma giunge alla seconda istruzione
                             ' NEXT x , il programma ritorna alla seconda istruzione FOR x= che
                             ' incrementa x di 3 pixel . Il ciclo si ripete con incrementi di (3) per arrestarsi
                             ' quando il valore di x ha raggiunto il valore (459)  con la
                             ' ultimazione della  prima punteggiata orizzontale
                             ' Il programma passa  quindi alla seconda istruzione NEXT y che lo invia
                             ' al secondo FORy=  per l'impostazione della punteggiata successiva

PSET ( x, y ),7   ' comanda il posizionamento dei punti per il tracciamento delle punteggiate orizzontali
                  ' in base alla seconda coppia delle istruzioni FOR , (colore della punteggiata =  bianco)

NEXT  x ' comanda il programma al ritorno automatico alla  seconda istruzione FOR  x =

NEXT y ' comanda il programma al ritorno automatico alla seconda  istruzione FOR y =

' LINE ( 0 , 160 ) - ( 460 , 160 )  ' comanda il tracciamento dell'asse delle ascisse (asse X)
                                    ' per coordinate a 2 e a 4 quadranti (colore = bianco luminoso)

' LINE ( 0 , 320 ) - ( 460 , 320 )  ' comanda il tracciamento dell'asse delle ascisse (asse X)
                                    ' per coordinate ad  1 quadrante (colore = bianco luminoso)

'LINE ( 230 , 0 ) - ( 230 , 320 )  ' comanda il tracciamento dell'asse delle ordinate ( asse Y)
                                   ' per coordinate a 4 quadranti  (colore = bianco luminoso)

'LINE ( 0 , 0 ) - ( 0 , 320 )       ' comanda il tracciamento dell'asse delle ordinate ( asse Y)
                                   ' per coordinate ad  1  o 2 quadranti (colore = bianco luminoso)
```

Si osservi che le ultime 4 istruzioni sono "bloccate "dal simbolo ' e compaiono come se fossero quattro commenti; le due istruzioni che saranno necessarie in base al tipo di funzione da presentare dovranno essere "sbloccate" togliendo il simbolo ' .
Se si deve presentare una funzione ordinaria, si devono sbloccare la prima la terza istruzione, in questo modo il sistema di assi cartesiani viene diviso nei 4 quadranti tradizionali con l'asse Y al centro del reticolo. In questo caso l'asse X si estende a sinistra rispetto all'asse Y per i valori

negativi delle ascisse, ed a destra dell'asse Y per i valori positivi delle ascisse; ogni ascissa è suddivisa in 10 intervalli .
Se invece si deve presentare una funzione pari, funzione che ha identico profilo sia a destra che a sinistra dell'ascissa x = 0, è conveniente sbloccare la prima e l'ultima istruzione al fine di utilizzare tutto lo spazio disponibile per il grafico realizzando un sistema di assi cartesiani a 2 quadranti; infatti in questo caso l'asse Y viene posto all'estrema sinistra del reticolo e la funzione può essere tracciata, in una delle sue due parti uguali, con il doppio dello spazio rispetto al caso precedente, con una sola ascissa per i valori positivi di X; l'ascissa è divisa in 20 intervalli.
Se infine si deve presentare una funzione sempre positiva che si sviluppa soltanto per valori di X>0 risulta conveniente formare un sistema di assi cartesiani ad un solo quadrante sbloccando la seconda e la quarta istruzione; in questo caso l'asse Y viene posto all'estrema sinistra del reticolo e l'asse X in fondo al reticolo e si dispone di tutto il reticolo per il tracciamento della funzione; si ha che l'ascissa e l'ordinata sono divise in 20 intervalli.
Premendo F5, se si sono sbloccate ad esempio la prima e la terza istruzione, si ha la presentazione del sistema di assi cartesiani a 4 quadranti come mostrato in figura 2 .
Per consentire una migliore visione dei grafici nel testo questi sono stampati con tratto nero su fondo bianco, mentre gli stessi compaiono sullo schermo a tratto bianco o colorato su sfondo nero.

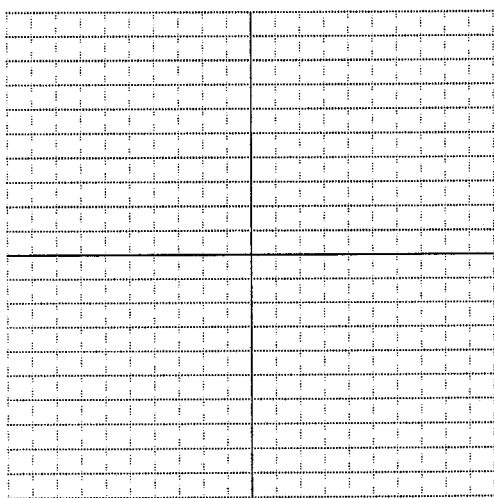



Figura 2
Sistema di assi cartesiani
a 4 quadranti con reticolo

Il programma che abbiamo ora illustrato inizia con l'istruzione grafica principale SCREEN 9.
Questa prepara la routine ad accogliere le istruzioni grafiche: PSET (x , y) e LINE (a , b)-(c , d).
E' necessario rimarcare che nel prosieguo del testo numerosi programmi richiederanno l'impiego del sistema di assi cartesiani con reticolo senza il quale, mancando ad essi l'istruzione principale SCREEN 9, non potranno "girare" da soli.

3.15 Il tracciamento dei grafici delle funzioni matematiche

L'istruzione PSET (x , y) si evidenzia in tutta la sua potenzialità nell'operazione di tracciamento dei grafici delle funzioni matematiche .
Le traiettorie dei punti governate da PSET(x , y) seguono i valori delle coordinate x; y anche quando queste sono a loro volta governate da funzioni di una o più variabili indipendenti.

L' unico vincolo alle traiettorie è determinato dalle dimensioni del reticolo fissato per il sistema di assi cartesiani; per le nostre applicazioni abbiamo stabilito, nel paragrafo 3.14, le dimensioni del reticolo in 460 pixel per le ascisse e 320 pixel per le ordinate; ciò implica che il campo di variabilità della coordinata x che comanda la traiettoria orizzontale del punto non deve essere superiore a 460 pixel e che il campo di variabilità della coordinata y che comanda la traiettoria verticale non deve essere superiore a 320 pixel.
A questo punto è necessario fare alcune precisazioni in merito ai campi di variabilità delle coordinate ed ai rapporti con le funzioni da tracciare:

Sul campo di variabilità della x
Se il sistema di assi cartesiani è stato diviso in 4 quadranti il campo di variabilità della x impegna 230 pixel per il semicampo dei valori negativi e 230 pixel per il semicampo dei valori positivi.
Se il sistema di assi cartesiani è stato diviso in 1 o 2 quadranti il campo unico di variabilità della x impegna 460 pixel.

Sul campo di variabilità della y
Se il sistema di assi cartesiani è stato diviso in 2 o 4 quadranti il campo di variabilità della y impegna 160 pixel per il semicampo dei valori negativi e 160 pixel per il semicampo dei valori positivi.
Se invece il sistema di assi cartesiani è stato diviso in 1 solo quadrante il campo di variabilità della y impegna 320 pixel.

Sulle funzioni da tracciare
Per il tracciamento delle funzioni si devono considerare separatamente la variabile indipendente, che agisce sulla coordinata x del punto, e la variabile dipendente che agisce sulla coordinata y; per entrambe le variabili è necessario stabilire il valore del rispettivo campo di variabilità:
-nel caso che questo non contenga il valore zero se ne considera in blocco tutta l'estensione
-nel caso che il campo contenga il valore zero si considerano le due parti che lo costituiscono, il semicampo positivo e il semicampo negativo

Entrambe le variabili devono essere rapportate alle scale disponibili mediante la moltiplicazione con adatti coefficienti:

La variabile indipendente per sistema di assi cartesiani a 4 quadranti
-si rapporta la variabile indipendente con la scala del reticolo moltiplicando x per il coefficiente k, dove k = 230 pixel / (valore del semicampo x)
-si ottiene il posizionamento del punto all'intersezione degli assi, quando x = 0, sommando 230 pixel a k * x
-la variabile indipendente dell'istruzione PSET assume pertanto l'espressione **230 + k * x**

La variabile indipendente per sistema di assi cartesiani ad 1 e a 2 quadranti
-si rapporta la variabile indipendente con la scala del reticolo moltiplicando x per il coefficiente k1, dove k1 = 460 pixel / (valore del campo x)
-il posizionamento del punto all'intersezione degli assi quando x = 0 è automatico
-la variabile indipendente dell'istruzione PSET assume l'espressione **k1 * x**

La variabile dipendente per i sistemi di assi cartesiani a 2 e 4 quadranti
-si rapporta la variabile dipendente con la scala del reticolo moltiplicando y per il coefficiente k2,

dove   k2 = 160 pixel / (valore del semicampo y)
-si ottiene il posizionamento del punto all'intersezione degli assi, quando y = 0, sottraendo y * k2 da 160 pixel
-la variabile dipendente dell'istruzione PSET assume pertanto l'espressione **160 - y * k2**

La variabile dipendente per i sistemi di assi cartesiani ad 1 quadrante
-si rapporta la variabile dipendente con la scala del reticolo moltiplicando y per il coefficiente k3, dove   k3 = 320 pixel / (valore del campo y)
-si ottiene il posizionamento all'intersezione degli assi sottraendo y * k3 da 320 pixel
-la variabile dipendente dell'istruzione PSET assume pertanto l'espressione **320 - y * k3**

Sulla base di quanto sopra specificato riassumiamo infine le istruzioni PSET necessarie per il tracciamento dei grafici delle funzioni matematiche nel sistema di assi cartesiani che abbiamo costruito:
per 4 quadranti   **PSET (230 + k * x , 160 - k2 * y)**
per 2 quadranti   **PSET (k1 * x , 160 - k2 * y)**
per 1 quadrante   **PSET (k1 * x , 320 - k3 * y)**

3.16 Esercitazione grafica n° 9 (tracciamento della funzione Sen x)

Disponendo del programma per la presentazione del sistema di assi cartesiani, sviluppato nel paragrafo 3.14, e delle nuove espressioni dell'istruzione PSET, elaborate nel paragrafo 3.15, vediamo come tracciare la curva della funzione y = Sen x:
1° si stabilisce il campo di variabilità della x, ad esempio, tra - 1.57 e +1.57 radianti
2° si fissa l'incremento dei passi della variabile indipendente in .00314 radianti
3° in base alla natura della funzione (funzione dispari) si sceglie la presentazione cartesiana a 4 quadranti "sbloccando" la prima e la terza istruzione del programma di paragrafo 3.14
4° si riporta l'istruzione PSET relativa ai 4 quadranti **PSET (230 + k * x , 160 - k2 * y)** nella quale x coincide con la variabile indipendente della funzione e y rappresenta la funzione stessa y = Sen x
5° si determina il coefficiente k in base al semicampo di variabilità fissato per la x in 1.57
  k = 230 / 1.57 = 146.49
6° si determina il coefficiente k2 in base al semicampo di variabilità previsto per y, che per la funzione y = Sen x è uguale ad 1
  k2 = 160 / 1 = 160
7° sulla base dei punti 4°, 5°, 6°, si completa, in simbologia Qbasic, l'istruzione finale che ci permette di tracciare la nostra funzione
PSET (230 + 146.49 * x , 160 - 160 * SIN (x))
8° si completa il programma del paragrafo 3.14 con l'aiuto dei punti 1° 2° e 7°:

```
' PROGRAMMA PER LA COSTRUZIONE DI UN SISTEMA DI ASSI CARTESIANI

SCREEN 9

FOR x = 0 TO 460 STEP 23 ' impone che l'ascissa (x) inizi il calcolo assumendo il valore (0)
                         ' resta poi in attesa che l'istruzione successiva costruisca la prima
                         ' punteggiata. Dopo che la prima punteggiata è stata tracciata
                         ' il programma giunge alla prima istruzione NEXT x che riporta la
                         ' esecuzione su il primo FOR x = che incrementa x di 23 pixel,
                         ' il processo si ripete fino al valore di x = 460 per il tracciamento
                         ' dell'ultima punteggiata verticale.
FOR y = 0 TO 320 STEP 2 ' impone che l'ordinata (y) inizi il calcolo assumendo il valore (0)     &
```

' quando l'esecuzione del programma giunge alla prima istruzione
' **NEXT y**, il programma ritorna alla prima istruzione **FOR y=** che incrementa y
' di 2 pixel . Il ciclo si ripete con incrementi di (2) per arrestarsi
' quando il valore di y ha raggiunto il valore (320) con la
' ultimazione della prima punteggiata verticale
' Il programma passa quindi alla prima istruzione **NEXT x** che lo invia
' al primo **FORx=** per l'impostazione della punteggiata successiva

PSET (x, y),7 ' comanda il posizionamento dei punti per il tracciamento delle punteggiate verticali
' in base alla prima coppia di istruzioni **FOR** (colore della punteggiata = bianco)

NEXT y ' comanda il programma al ritorno automatico alla prima istruzione **FOR y =**

NEXT x ' comanda il programma al ritorno automatico alla prima istruzione **FOR x =**

FOR y = 0 TO 320 STEP 16 ' impone che l'ordinata (y) inizi il calcolo assumendo il valore (0)
' resta poi in attesa che l'istruzione successiva costruisca la prima
' punteggiata. Dopo che la prima punteggiata è stata tracciata
' il programma giunge alla seconda istruzione **NEXT y** che riporta
' l'esecuzione sul secondo **FOR y =** che incrementa y di 16 pixel,
' il processo si ripete fino al valore di y = 320 per il tracciamento
' dell'ultima punteggiata orizzontale.

FOR x = 0 TO 460 STEP 3 ' impone che l'ascissa (x) inizi il calcolo assumendo il valore (0)
' quando l'esecuzione del programma giunge alla seconda istruzione
' **NEXT x** , il programma ritorna alla seconda istruzione **FOR x=** che
' incrementa x di 3 pixel . Il ciclo si ripete con incrementi di (3) per arrestarsi
' quando il valore di x ha raggiunto il valore (459) con la
' ultimazione della prima punteggiata orizzontale
' Il programma passa quindi alla seconda istruzione **NEXT y** che lo invia
' al secondo **FORy=** per l'impostazione della punteggiata successiva

PSET (x, y),7 ' comanda il posizionamento dei punti per il tracciamento delle punteggiate orizzontali
' in base alla seconda coppia delle istruzioni **FOR** , (colore della punteggiata = bianco)

NEXT x ' comanda il programma al ritorno automatico alla seconda istruzione **FOR** x =

NEXT y ' comanda il programma al ritorno automatico alla seconda istruzione **FOR** y=

LINE (0 , 160) - (460 , 160) ' comanda il tracciamento dell'asse delle ascisse (asse X)
' per coordinate a 2 e a 4 quadranti (colore = bianco luminoso)

'LINE (0 , 320) - (460 , 320) ' comanda il tracciamento dell'asse delle ascisse (asse X)
' per coordinate ad 1 quadrante (colore = bianco luminoso)

LINE (230 , 0) - (230 , 320) ' comanda il tracciamento dell'asse delle ordinate (asse Y)
' per coordinate a 4 quadranti (colore = bianco luminoso)

'LINE (0 , 0) - (0 , 320) ' comanda il tracciamento dell'asse delle ordinate (asse Y)
' per coordinate ad 1 o 2 quadranti (colore = bianco luminoso)

' PROGRAMMA PER IL TRACCIAMENTO DELLA FUNZIONE Y = Sen x

FOR x = -1.57 TO 1.57 STEP .00314 ' campo di variabilità ed incremento della x (punto 1° e 2°)

PSET (230 + 146.49 * x , 160 - 160 * SIN (x)) ' istruzione elaborata al punto 7°

NEXT x ' comanda il programma al ritorno automatico all'istruzione FOR x = -1.57 ecc.

F5

si ha la presentazione grafica della funzione y = Sen x come mostrato in figura 3.

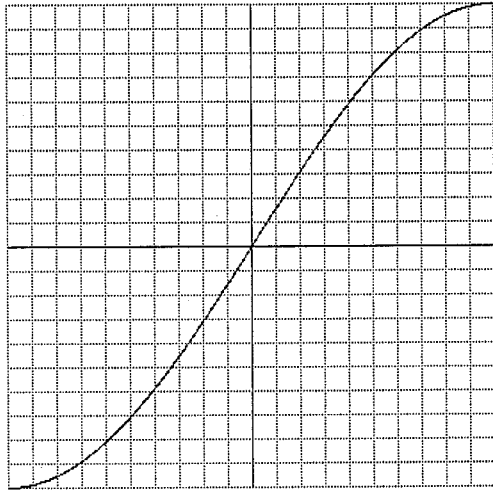



Figura 3
Grafico della funzione Sen x
Campo di variabilità della x :
da -1.57 a +1.57 radianti
Scala asse x = .157 rad. / div.
Scala asse y = .1 / div.

3.17 Esercitazione grafica n° 10 (tracciamento della funzione Sen x / x)

Ci proponiamo di tracciare la curva della funzione

$$y = \frac{\text{Sen } x}{x}$$

1° si stabilisce il campo di variabilità della x, ad esempio, tra .00001 e 30 radianti (si osservi che nel campo di variabilità della x è stato escluso lo zero per non generare forme indeterminate)
2° si fissa l'incremento dei passi della variabile indipendente in .01 radianti
3° in base alla natura della funzione (funzione pari con la y definita nei due semicampi negativo e positivo) si sceglie la presentazione cartesiana a 2 quadranti "sbloccando" la prima e la quarta istruzione del programma di paragrafo 3.14
4° si riporta l'istruzione PSET relativa ai 2 quadranti **PSET (k1 * x , 160 - k2 * y)**
5° si determina il coefficiente k1 in base al campo di variabilità fissato per la x in 30 radianti
k1 = 460 / 30 = 15.33
6° si determina il coefficiente k2 in base al semicampo di variabilità previsto per y, che per la funzione y = Sen x /x è uguale ad 1: k2 = 160 / 1 = 160
7° sulla base dei punti 4°, 5°, 6°, si completa, in simbologia Qbasic, l'istruzione finale che rende possibile il tracciamento della nostra funzione, si aggiunge il colore giallo:

```
PSET ( 15.33 * x , 160 - 160 * SIN ( x ) / x ), 14
```

8° si completa il programma del paragrafo 3.14 con l'aiuto dei punti 1° 2° e 7°:

Da questo esercizio in poi, per semplificare il testo, non ripeteremo completamente il programma per il tracciamento delle coordinate ma ci limiteremo a riscrivere soltanto le due istruzioni "sbloccate" e la routine necessaria ad accogliere la funzione da tracciare. Sarà cura del lettore, che si accinge a compilare il programma, richiamare da memoria o digitare la parte omessa .Senza l'istruzione SCREEN 9 di tale parte i programmi grafici non possono girare.

```
LINE ( 0 , 160 ) - ( 460 , 160 )  ' comanda il tracciamento dell'asse delle ascisse (asse X)
                                  ' per coordinate a 2 e a 4 quadranti (colore = bianco luminoso)

LINE ( 0 , 0 ) - ( 0 , 320 )  ' comanda il tracciamento dell'asse delle ordinate ( asse Y)
                              ' per coordinate ad 1 o 2 quadranti (colore = bianco luminoso)

FOR x = .00001 TO 30 STEP .01  ' campo di variabilità ed incremento della x (punto 1° e 2°)

PSET ( 15.33 * x , 160 - 160 * SIN ( x ) / x ), 14  ' istruzione elaborata al punto 7°

NEXT x  ' comanda il programma al ritorno automatico all'istruzione FOR x = .00001 ecc.
```

premuto F5 si ha la presentazione della funzione come mostrato in figura 4.

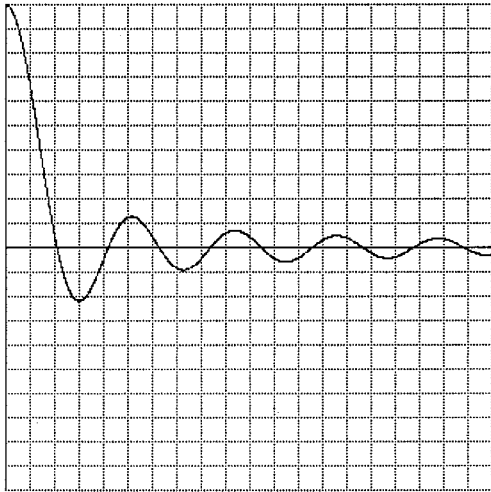



Figura 4
Grafico della funzione Sen x / x
Campo di variabilità della x :
da .00001 a 30 radianti
Scala asse x = 1.5 rad. /div.
Scala asse y = .1 / div.

3.18 Esercitazione grafica n° 11 (tracciamento della funzione gaussiana)

Proponiamoci di tracciare il grafico della funzione gaussiana

$$y = e^{-a x^2} \qquad \text{per } a = .01$$

la funzione è pari e si estende soltanto nel campo positivo della y; è opportuno perciò, per una più ampia presentazione del tracciato, impiegare il sistema di assi cartesiani ad 1 solo quadrante:
1° si stabilisce il campo di variabilità della x, ad esempio, tra 0 e +20

2° si fissa l'incremento dei passi della variabile indipendente in .01
3° in base alla natura della funzione si sceglie la presentazione cartesiana ad 1 quadrante "sbloccando" la seconda e la quarta istruzione del programma di paragrafo 3.14
4° si riporta l'istruzione PSET relativa ad 1 quadrante **PSET (k1 * x , 320 - k3 * y)**
5° si determina il coefficiente k1 in base al campo di variabilità fissato per la x in 20
k 1 = 460 / 20 = 23
6° si determina il coefficiente k3 in base al campo di variabilità previsto per y, che per la funzione gaussiana è uguale ad 1: k3 = 320 / 1 = 320
7° sulla base dei punti 4°, 5°, 6°, si completa, in simbologia Qbasic, l'istruzione finale che ci permette di tracciare la nostra funzione, si aggiunge il colore giallo:

PSET (23 * x , 320 - 320 * EXP (-a * x ^ 2)), 14

8° si completa il programma del paragrafo 3.14 con l'aiuto dei punti 1° 2° e 7°

```
LINE (0 , 320) - ( 460 , 320 ) ' comanda il tracciamento dell'asse delle ascisse (asse X) per 1 quadrante

LINE ( 0 , 0 ) - ( 0 , 320 ) ' comanda il tracciamento dell'asse delle ordinate ( asse Y) per 1 quadrante

a = .01 ' parametro della gaussiana fissato a .01

FOR x= 0 TO 20 STEP .01 ' campo di variabilità ed incremento della x (punto 1° e 2°)

PSET ( 23 * x , 320 - 320 * EXP ( -a * x ^ 2 ) ) , 14 ' istruzione elaborata al punto 7°

NEXT x ' comanda il programma al ritorno automatico all'istruzione FOR x = 0 ...
```

pigiando F5 si ha la presentazione della curva di figura 5

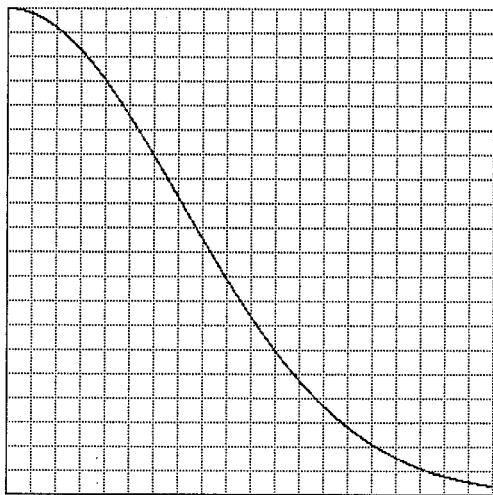



Figura 5
Grafico della funzione gaussiana
Campo di variabilità della x:
da 0 a 20
Scala asse x = 1 / div.
Scala asse y = .05 / div.

3.19 La presentazione contemporanea di più funzioni

E' facile estendere il procedimento per il tracciamento della curva di una funzione al tracciamento di due o più funzioni contemporaneamente. Naturalmente in questo caso non si può scegliere il sistema di assi cartesiani che ottimizza una delle funzioni ma si deve optare per il sistema di assi che favorisce il tracciamento della funzione che necessita del numero più elevato di quadranti anche se ciò riduce la dimensione di presentazione delle altre curve.
E' utile in tali applicazioni tracciare le curve a colori diversi per poterle poi distinguere meglio l'una dall'altra. Una applicazione pratica di questo nuovo procedimento è riportata nel successivo paragrafo.

3.20 Esercitazione grafica n° 12 (il tracciamento di più funzioni)

Con questa esercitazione mostriamo la semplicità e l'importanza della metodologia di tracciamento contemporaneo di più funzioni; per comodità scegliamo le funzioni esaminate nei paragrafi 3.16; 3.17; 3.18 da definire in campi di variabilità della x compatibili tra loro:

$y = \text{Sen } x$

$y = \text{Sen } x / x$

$y = e^{-a x^2}$

Per quanto indicato nel paragrafo 3.19 dobbiamo scegliere il sistema di assi cartesiani a 4 quadranti necessario per la presentazione della funzione y = Sen x anche se ciò costringe i grafici delle altre due funzioni ad essere tracciati in superfici ristrette; adottando i 4 quadranti, inoltre, vengono tracciate le parti simmetriche delle due curve pari che negli esercizi precedenti non sono state presentate, ed infine per diversificare i tracciati delle tre funzioni è opportuno colorare le curve in modo diverso.

Ciò premesso dobbiamo riproporre i termini di impostazione del lavoro come segue:

-per la funzione y = Sen x

1° si stabilisce il campo di variabilità della x, ad esempio, tra - 10 e +10 radianti
2° si fissa l'incremento dei passi della variabile indipendente in .01 radianti
3° in base alla natura della funzione (funzione dispari) si sceglie la presentazione cartesiana a 4 quadranti "sbloccando" la prima e la terza istruzione del programma di paragrafo 3.14
4° si riporta l'istruzione PSET relativa ai 4 quadranti **PSET (230 + k * x , 160 - k2 * y)**
5° si determina il coefficiente k in base al semicampo di variabilità fissato per la x in 10 radianti
k = 230 / 10 = 23
6° si determina il coefficiente k2 in base al semicampo di variabilità previsto per y, che per la funzione y = Sen x è uguale ad 1
k2 = 160 / 1 = 160
7° sulla base dei punti 4°, 5°, 6°, si completa, in simbologia Qbasic, l'istruzione finale che ci consente di tracciare la nostra funzione, si aggiunge il colore verde:

```
PSET ( 230 + 23 * x , 160 - 160 * SIN ( x ) ) ,2
```

-per la funzione Sen x / x

1° si stabilisce il campo di variabilità della x, per uniformità con la prima funzione, tra -10 e + 10 radianti (si osservi che nel campo di variabilità della x l'esclusione dello zero è automatica grazie all'entità dell'incremento fissato)
2° si fissa l'incremento dei passi della variabile indipendente come per la prima funzione in .01 radianti
3° indipendentemente dal tipo di funzione il sistema cartesiano è vincolato dalla prima funzione in 4 quadranti
4° si riporta l'istruzione PSET relativa ai 4 quadranti **PSET (230 + k * x , 160 - k2 * y)**
5° si determina il coefficiente k in base al semicampo di variabilità fissato per la x in 10 radianti
k = 230 / 10 = 23
6° si determina il coefficiente k2 in base al semicampo di variabilità previsto per y, che per la funzione y = Sen x /x è uguale ad 1:
k2 = 160 / 1 = 160
7° sulla base dei punti 4°, 5°, 6°, si completa, in simbologia Qbasic, l'istruzione finale per tracciare la nostra funzione, si aggiunge il colore giallo:

PSET (230 + 23 * x , 160 - 160 * SIN (x) / x) , 14

-per la funzione gaussiana (si fissa a = .03 per una più chiara dimostrazione didattica)

1° si stabilisce il campo di variabilità della x in conformità con le altre funzioni da -10 a +10
2° si fissa l'incremento dei passi della variabile indipendente come per le altre funzioni .01
3° il sistema di assi cartesiani è già fissato in precedenza in 4 quadranti
4° si riporta l'istruzione PSET relativa ai 4 quadranti **PSET (230 + k * x , 160 - k2 * y)**
5° si determina il coefficiente k in base al semicampo di variabilità fissato per la x in 10
k = 230 / 10 = 23
6° si determina il coefficiente k2 in base al campo di variabilità previsto per y, che per la funzione gaussiana è uguale ad 1:
k2 = 160 / 1 = 160
7° sulla base dei punti 4°, 5°, 6°, si completa, in simbologia Qbasic, l'istruzione finale che ci permette di tracciare la nostra funzione, colore bianco:

PSET (230 + 23 * x , 160 - 160 * EXP (-a * x ^ 2))

Con le tre istruzioni che abbiamo costruito compiliamo il programma voluto:

LINE (0 , 160) - (460 , 160) ' comanda il tracciamento dell'asse delle ascisse (asse X) coordinate a 4 quadranti

LINE (230 , 0) - (230 , 320) ' comanda il tracciamento dell'asse delle ordinate (asse Y) coordinate a 4 quadranti

FOR x = - 10 TO 10 STEP .01 ' campo di variabilità ed incremento della x (punti 1° e 2°)

PSET (230 + 23 * x , 160 - 160 * SIN (x)), 2 ' gestisce il tracciamento della curva Seno (colore verde)

PSET (230 + 23 * x , 160 - 160 * SIN (x) / x), 14 ' gestisce il tracciamento della curva Sen x/x (colore giallo)

a = .03 ' parametro della gaussiana

PSET (230 + 23 * x , 160 - 160 * EXP (-a * x ^ 2)) ' gestisce il tracciamento curva gaussiana

NEXT x ' rimanda il programma al ritorno automatico all'istruzione FOR x = - 10 ecc.

F5

si ha la presentazione contemporanea delle tre curve come mostrato in figura 6

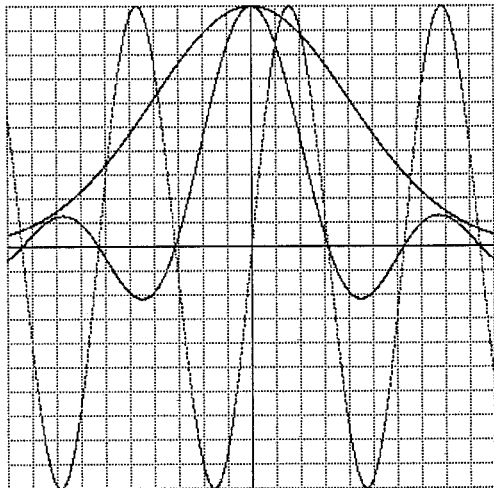



Figura 6
Grafico di tre funzioni
Campo di variabilità della x :
da -10 a +10 radianti
Scala asse x = 1 rad./div.
Scala asse y = .1 / div.

3.21 Esercitazione grafica n° 13 (tracciamento della funzione tangente)

Per introdurre il lettore all'impiego dell'istruzione PSET sono state usate, per semplicità, funzioni elementari il cui campo di variabilità di y si estende sempre tra 0 e +/- 1.
Esercitiamoci ora con una funzione elementare che ha un campo di variabilità di y diverso da +/-1; sia questa la funzione trigonometrica

$$Y = \text{Tang } x$$

della quale vogliamo tracciare il grafico rappresentativo nel campo di variabilità della x compreso tra - 1.5 e + 1 radianti.
Si deve osservare che nella compilazione dell'istruzione PSET dobbiamo impiegare il valore più grande (in valore assoluto) degli estremi che definiscono i semicampi di variabilità sia della x che della y, per far sì che tutta la funzione possa essere rappresentata nel grafico senza troncamenti.
Definito il campo di variabilità della x è necessario valutare il corrispondente campo di variabilità della y per poi procedere alla compilazione del programma grafico; per fare ciò dobbiamo iniziare compilando un piccolo programma sulla base dei valori assegnati ad x:

```
CLS ' pulizia dello schermo

FOR x = - 1.5 TO 1 STEP .5 ' impone che la variabile indipendente (x) inizi il calcolo assumendo
                           ' il valore (-1. 5) , quando l'esecuzione del programma giunge alla istruzione
                           ' NEXT il programma ritorna all'istruzione FOR che incrementa x del        &
```

```
                          ' valore (. 5 ) . Il ciclo si ripete con incrementi di (. 5) per arrestarsi
                          ' quando il valore di x ha raggiunto il valore (1)
Y = TAN ( x ) ' funzione trigonometrica da computare

PRINT "Y = " ; Y  ' comando visualizzazione dati variabile dipendente

NEXT x ' comanda il programma al ritorno automatico all'istruzione FOR..
```

```
            F5
Y = -14.10142
Y = -1.557408
Y = -.5463025
Y = 0
Y = .5463025
Y = 1.557408
            F5
```

Lo sviluppo del programma ci ha fornito i valori del campo di variabilità di Y:
da - 14.10142 a + 1.557408 . A questo punto il lettore potrebbe pensare che il calcolo che è stato sviluppato poteva limitarsi ai soli estremi del campo di variabilità della x; ciò è vero, in questo caso, dato che la funzione in esame, nel campo di variabilità assunto per la x, è crescente. La procedura seguita è però d'obbligo dato che, nella generalità dei casi, le funzioni possono essere sia crescenti, sia decrescenti, sia dotate di massimi e minimi; è perciò indispensabile un'indagine all'interno del campo di variabilità per stabilire i valori massimi che la variabile indipendente può eventualmente assumere in qualsiasi punto del campo.
Chiarito questo significativo aspetto dell'indagine procediamo ora alla compilazione dell'istruzione PSET:
1° si riporta il campo di variabilità della x: da -1.5 a 1 radianti
2° si fissa l'incremento dei passi della variabile indipendente in .001 radianti
3° in base alla natura della funzione (funzione dispari) si sceglie la presentazione cartesiana a 4 quadranti "sbloccando" la prima e la terza istruzione del programma di paragrafo 3.14
4° si riporta l'istruzione PSET relativa ai 4 quadranti **PSET (230 + k * x , 160 - k2 * y)**
5° si determina il coefficiente k in base al semicampo di variabilità più grande fissato per la x in -1.5 radianti k = 230 / 1.5 = 153.33 (il calcolo è eseguito sul valore assoluto di x)
6° si determina il coefficiente k2 in base al semicampo di variabilità più grande calcolato per y in - 14.10141; k2 = 160 / 14.10141 = 11.346 (il calcolo è eseguito sul valore assoluto di y)
7° sulla base dei punti 4°, 5°, 6°, si completa, in simbologia Qbasic, l'istruzione finale che ci consente di tracciare la nostra funzione, si aggiunge il colore verde:

PSET (230 + 153.33 * x , 160 - 11.346 * TAN (x)) ,2

8 ° si procede alla compilazione del programma:

```
LINE ( 0 , 160 ) - ( 460 , 160 )  ' comanda il tracciamento dell'asse delle ascisse (asse X)
                                   ' per coordinate a 2 e a 4 quadranti (colore = bianco luminoso)

LINE ( 230 , 0 ) - ( 230 , 320 ) ' comanda il tracciamento dell'asse delle ordinate ( asse Y)
                                  ' per coordinate a 4 quadranti (colore = bianco luminoso)

FOR x = - 1.5 TO 1 STEP .001  ' campo di variabilità ed incremento della x (punti 1° e 2°)

PSET ( 230 + 153.33 * x , 160 - 11.346 * TAN ( x ) ) ,2  ' gestisce il tracciamento della curva
                                                         ' Tangente (colore verde)
NEXT x ' invia all'istruzione FOR
```

F5

si ha la presentazione del grafico voluto mostrato in figura 7

Figura 7
Grafico della funzione Tang x
Campo di variabilità della x :
da -1.5 a + 1 radianti
Scala asse x = .15 rad. /div.
Scala asse y = 1.41 / div.

A conclusione di questa esercitazione è necessario far osservare che sono stati scelti due passi diversi di incremento della x: .5 radianti per il calcolo degli estremi del campo di variabilità della y e .001 radianti per il tracciamento del grafico della funzione. Ciò dipende da precise esigenze:

-per il calcolo degli estremi del campo di variabilità di y è sufficiente che l'incremento di x permetta di coprire il campo della sua variabilità con precisione particolare agli estremi senza necessità di calcolare molti valori di y all'interno del campo salvo che all'interno di questo non sia denunciato un massimo.

-per il tracciamento della curva della funzione è necessario disporre di molti valori di y in modo da rendere la presentazione della stessa uniforme e non punteggiata; per questa ragione, assumendo l'incremento di .001 radianti, si calcolano (|-1.5 | + 1) / .001 = 2500 punti di y che sono di gran lunga superiori al numero massimo dei 460 pixel dell'asse x.

3.22 L'istruzione LOCATE

E' di notevole interesse, per i sistemi grafici, l'istruzione **LOCATE** che permette di aggiungere delle scritte nel reticolo del sistema di assi cartesiani. Ricordiamo che la modalità di schermo SCREEN 9 consente, come accennato al paragrafo 3.2, la scrittura di 80 caratteri in orizzontale e 24 caratteri in verticale, caratteri che possono essere evidenziati con la nuova istruzione.
A fianco del reticolo si possono scrivere, per ricordarle, informazioni varie riguardanti le funzioni tracciate e tutto ciò che l'operatore ritenga necessario
Per utilizzare l'istruzione LOCATE la si deve completare, sia con le "coordinate di carattere", sia

con i caratteri che devono essere scritti a partire da dette coordinate.
Le "coordinate di carattere" contengono il numero di posizione verticale del carattere, che può essere scelto tra 1 e 24, ed il numero di posizione orizzontale che può essere scelto tra 1 e 80 (se si sceglie il valore 80 un solo carattere può essere scritto).
Se ad esempio vogliamo scrivere i valori attribuiti alle divisioni del reticolo, orizzontale (.55) e verticale (1.34), collocando tali scritte nell'angolo inferiore destro dello schermo si compilano due istruzioni nel seguente modo:

per l'asse Y: LOCATE 23 , 66 : PRINT "Y-Div. = 1.34"
-il numero 23 definisce la posizione della riga di scrittura alla ventitreesima fila partendo dall'alto dello schermo
-il numero 66 definisce l'inizio della scrittura al sessantaseiesimo carattere partendo dal lato sinistro dello schermo
-i caratteri Y-Div. = 1.34 rappresentano ciò che viene scritto nella posizione definita dalle coordinate di carattere

per l'asse X: LOCATE 22 , 66 : PRINT "X-Div. = .55"
-il numero 22 definisce la posizione della riga di scrittura alla ventiduesima fila partendo dall'alto dello schermo
-il numero 66 definisce l'inizio della scrittura al sessantaseiesimo carattere partendo dal lato sinistro dello schermo
-i caratteri X-Div. = .55 rappresentano ciò che viene scritto nella posizione definita dalle coordinate di carattere

Per non incorrere in errori nell'impiego delle istruzioni LOCATE è consigliabile inserirle come ultime nel contesto della stesura del programma grafico; ciò è mostrato nell'esercitazione del paragrafo 3.23.

3.23 Esercitazione grafica n° 14 (l'impiego delle istruzioni LOCATE)

Al fine di esercitarci sull'impiego delle istruzioni LOCATE prendiamo in considerazione l'applicazione grafica n ° 13; proponiamoci l'inserimento dei valori di scala del reticolo che andiamo a valutare:

per l'asse X ogni intervallo vale | -1.5 | / 10 intervalli = .15
per l'asse Y ogni intervallo vale | -14.10141 | / 10 intervalli = 1.41

con questi numeri modifichiamo le istruzioni LOCATE, illustrate nel paragrafo precedente, con i valori da attribuire agli intervalli del reticolo, lasciamo invece inalterate le posizioni delle scritte che sono state scelte per non interferire con il grafico; collochiamo le nuove istruzioni alla fine del programma grafico n° 12 che riportiamo:

```
LINE ( 0 , 160) - ( 460 , 160 )  ' comanda il tracciamento dell'asse delle ascisse (asse X)
                                 ' per coordinate a 2 e a 4 quadranti (colore = bianco luminoso)

LINE ( 230 , 0 ) - ( 230 , 320 )  ' comanda il tracciamento dell'asse delle ordinate ( asse Y)
                                  ' per coordinate a 4 quadranti (colore = bianco luminoso)

FOR x = - 1.5 TO 1 STEP .001 ' campo di variabilità ed incremento della x (punti 1° e 2°)

PSET ( 230 + 153.33 * x , 160 - 11.346 * TAN ( x ) ) ,2 ' gestisce il tracciamento della curva
                                                        ' Tangente (colore verde)
NEXT x ' invia all'istruzione FOR                                                  &
```

```
LOCATE 23 , 66 : PRINT "Y- Div. = 1.41"     ' produce la scritta    Y-Div. = 1.41  nell'angolo
                                             ' basso a destra dello schermo

LOCATE 22 , 66 : PRINT "X- Div. = .15"      ' produce la scritta    X-Div: = .15  nell'angolo
                                             ' basso a destra dello schermo
```

F5

si ha la presentazione del grafico con le indicazioni del valore attribuito agli intervalli verticali e orizzontali del reticolo.

F5

3.24 L'istruzione PSET per funzioni composte

Le esercitazioni che abbiamo svolto nei paragrafi precedenti sono state volutamente indirizzate al tracciamento dei grafici delle funzioni elementari per mostrare come le funzioni matematiche possano diventare parte integrante dell'istruzione PSET. In effetti ciò è possibile anche per funzioni composte da più funzioni elementari, ma se queste sono molto complicate le istruzioni PSET sono compilabili con difficoltà. In tali casi si adotta un metodo semplice che dà modo di scrivere separatamente la funzione composta, di valutarne il campo di variabilità e di compilare l'istruzione PSET mediante un solo simbolo al posto della funzione della quale si deve tracciare la curva.
Il metodo consiste nell'impiego della scrittura di base dell'istruzione PSET che come ricordiamo ha la forma:

PSET (x , y)

che, per non creare confusioni con la generica nomenclatura delle variabili indipendenti e dipendenti delle funzioni da tracciare, può essere scritta:

PSET (a , b)

dove (a) e (b) hanno sostituito rispettivamente i caratteri x ed y mantenendo inalterato il loro ruolo di variabili di posizione.
A questo punto l'istruzione PSET (a , b), semplicemente come è scritta, può essere inserita nel contesto del programma grafico collegando opportunamente, mediante uguaglianze, le nuove variabili di posizione (a) , (b) alla funzione di cui si vuole tracciare il grafico.
Una sintesi di questo procedimento è mostrata nell'esercitazione grafica n° 15 .

3.25 Esercitazione grafica n° 15 (il tracciamento di una funzione composta)

Il presente esercizio ha lo scopo di mostrare l'impiego dell'istruzione PSET(a , b) per il tracciamento di una funzione composta; vediamo quindi come presentare il grafico della funzione:

$$Y = \frac{\ln(3 + 4x) + \mathrm{Sen}\ 2x}{5x + 1}$$

nel campo di variabilità di x compreso tra 0 e 2 radianti .

Similmente a quanto già fatto nel paragrafo 3.21 iniziamo a calcolare il campo di variabilità di Y:

In questo caso, prima di iniziare il calcolo, è necessario trasformare l'espressione ordinaria della funzione composta, mediante le corrispondenze simboliche, in una espressione in Qbasic; abbiamo pertanto

Y = (LOG (3 + 4 * x) + SIN (2 * x)) / (5 * x + 1)

che andiamo ad inserire nel seguente programma:

```
CLS  ' pulizia dello schermo

FOR x = 0 TO 2 STEP .5 ' impone che la variabile indipendente (x) inizi il calcolo assumendo
                       ' il valore ( 0 ) , quando l'esecuzione del programma giunge alla istruzione
                       ' NEXT il programma ritorna all'istruzione FOR che incrementa x del
                       ' valore (. 5 ) . Il ciclo si ripete con incrementi di (. 5) per arrestarsi
                       ' quando il valore di x ha raggiunto il valore (2)

Y = ( LOG ( 3 + 4 * x ) + SIN ( 2 * x ) ) / ( 5 * x + 1 ) ' funzione composta da computare

PRINT "Y = " ; Y ' comando visualizzazione dati variabile dipendente

NEXT x ' comanda il programma al ritorno automatico alla istruzione FOR...
```

```
            F5
Y = 1.098612
Y = .7002597
Y = .4758679
Y = .2750994
Y = .1491902
            F5
```

Lo sviluppo del programma ci ha fornito i valori del campo di variabilità di Y: da +1.098612 a + .1491902 nel quale si vede che la funzione è decrescente, procediamo ora alla compilazione dell'istruzione PSET (a , b) .
1° si riporta il campo di variabilità della x: da 0 a 2 radianti
2° si fissa l'incremento dei passi della variabile indipendente in .001 radianti
3° in base alla natura della funzione si sceglie la presentazione cartesiana ad 1 quadrante "sbloccando" la seconda e la quarta istruzione del programma di paragrafo 3.14
4° si riporta l'istruzione PSET relativa ad 1 quadrante **PSET (k1 * x , 320 - k3 * y)** si modifica l'istruzione in PSET (a , b) con le uguaglianze:
x = a   y = b scrivendo PSET (k1 * a , 320 - k3 * b)
5° si determina il coefficiente k1 in base al campo di variabilità fissato per la x = a in 2 radianti
k1 = 460 / 2 = 230
6° si determina il coefficiente k3 in base all'estremo del campo di variabilità calcolato per y = b
k3 = 320 / 1.09861 = 291.277
7° sulla base dei punti 4°, 5°, 6°, si completa la nuova istruzione PSET:
PSET (230 * a , 320 - 291.277 * b)
8 ° si procede alla compilazione del programma:

```
LINE (0 , 320) - ( 460 , 320 ) ' comanda il tracciamento dell'asse delle ascisse (asse X)
                                ' per coordinate ad 1 quadrante (colore = bianco luminoso)

LINE ( 0 , 0 ) - ( 0 , 320) ' comanda il tracciamento dell'asse delle ordinate ( asse Y)
                            ' per coordinate ad 1 o 2 quadranti (colore = bianco luminoso)       &
```

```
FOR x = 0 TO 2 STEP .001 ' campo di variabilità ed incremento della x (punti 1° e 2°)

Y = ( LOG ( 3 + 4 * x ) + SIN ( 2 * x ) ) / ( 5 * x + 1 ) ' calcolo della funzione composta

a = x   ' uguaglianza tra l' ascissa di posizione di PSET e la variabile indipendente della funzione
        ' nota bene- nell'eguaglianza il primo termine deve essere (a) ed il secondo termine x

b = Y  ' uguaglianza tra l'ordinata di posizione di PSET e la variabile dipendente della funzione
        ' nota bene- nell'eguaglianza il primo termine deve essere (b) ed il secondo termine y

PSET ( 230 * a , 320 - 291.277 * b ) ,2  ' gestisce il tracciamento della curva della funzione composta
                                          ' (colore verde)
NEXT x ' invia all'istruzione FOR

        F5
```

si ha la presentazione del grafico voluto mostrato in figura 8.

Si consiglia il lettore di ripetere questa esercitazione per diversi campi di variabilità della x ponendo molta attenzione a non uscire dal campo di variabilità naturale della funzione logaritmica che fa parte della funzione composta; attenzione inoltre a non scegliere campi di variabilità che contengano valori di x tali da azzerare il denominatore della funzione.

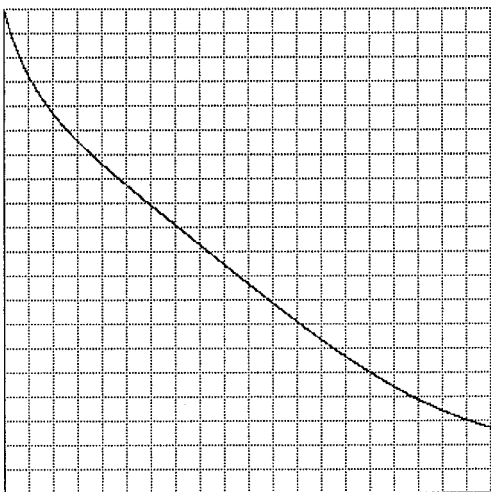



Figura 8
Grafico della funzione composta
Campo di variabilità della x :
da 0 a 2 radianti
Scala asse x = .1 rad. /div.
Scala asse y = .054 / div.

3.26 La calibrazione delle scale nei sistemi di assi cartesiani

In tutti gli esercizi grafici svolti in precedenza i coefficienti di scala impiegati sono stati dimensionati allo scopo di assegnare la massima estensione ai tracciati in modo da renderne migliore la visualizzazione. Con questo metodo gli intervalli in cui sono suddivisi gli assi cartesiani, tramite il reticolo, assumono valori che a volte non si prestano ad una lettura diretta sul grafico dei livelli che la funzione assume per dati valori di x; ciò può rendere poco agevole un eventuale esame visivo, se necessario, della curva e dei suoi particolari caratteristici. Per facilitare la lettura

diretta dei tracciati delle funzioni si può impiegare il metodo della calibrazione delle scale nei sistemi di assi cartesiani; l'applicazione di tale metodo è mostrata nel successivo paragrafo 3.27.

3.27 Esercitazione grafica n° 16 (come calibrare la scala di un tracciato di funzione)

Il presente esercizio è sviluppato per evidenziare il passaggio tra un sistema di assi cartesiani non calibrato (utilizzato per una migliore visualizzazione della curva rappresentativa di una funzione matematica) ed un sistema di assi cartesiani calibrato (utilizzato nei casi in cui sia necessario valutare visivamente i livelli della curva rappresentativa di una funzione).
Consideriamo la funzione

$$Y = \ln x$$

nel campo di variabilità della x compreso tra .7 e 4.201; iniziamo il lavoro per ottenere il tracciato del grafico in modo da visualizzare al meglio la curva della funzione:
calcoliamo il campo di variabilità della y

```
CLS ' pulizia dello schermo

FOR x = .7 TO 4.201 STEP .5 ' impone che la variabile indipendente (x) inizi il calcolo assumendo
                            ' il valore ( .7 ) , quando l'esecuzione del programma giunge alla istruzione
                            ' NEXT il programma ritorna all'istruzione FOR che incrementa x del
                            ' valore (. 5 ) . Il ciclo si ripete con incrementi di (. 5) per arrestarsi
                            ' quando il valore di x ha raggiunto il valore ( 4. 2)

Y = LOG ( x ) ' funzione da computare

PRINT "Y = " ; Y ' comando visualizzazione dati variabile dipendente

NEXT x
```

```
                F5
Y = -.356675
Y = .1823216
Y = .5306283
Y = .7884574
Y = .9932518
Y = 1.163151
Y= 1.308333
Y= 1.435084
                F5
```

dal calcolo si vede che la funzione è crescente e che il campo di variabilità della Y è compreso tra -.356675 e + 1.435084; seguiamo ora la procedura ordinaria:

1° si riporta il campo di variabilità della x: da .7 a 4.201
2° si fissa l'incremento dei passi della variabile indipendente in .001
3° in base alla natura della funzione si sceglie la presentazione cartesiana a 2 quadranti "sbloccando" la prima e la quarta istruzione del programma di paragrafo 3.14
4° si riporta l'istruzione PSET relativa ai 2 quadranti **PSET (k1 * x , 160 - k2 * y)**
5° si determina il coefficiente k1 in base al semicampo di variabilità più grande fissato per la x in 4.201 k1 = 460 / 4.201 = 109.5
6° si determina il coefficiente k2 in base al semicampo di variabilità più grande calcolato per y in 1.435084; k2 = 160 / 1.435084 = 111.491

7° sulla base dei punti 4°, 5°, 6°, si completa, in simbologia Qbasic, l'istruzione finale che ci consente di tracciare la nostra funzione, si aggiunge il colore verde:

PSET (109.5 * x , 160 - 111.491 * LOG (x)) ,2

8 ° si procede alla compilazione del programma:

```
LINE ( 0 , 160 ) - ( 460 , 160 ) ' comanda il tracciamento dell'asse delle ascisse (asse X)
                                 ' per coordinate a 2 e a 4 quadranti (colore = bianco luminoso)

LINE ( 0 , 0 ) - ( 0 , 320 ) ' comanda il tracciamento dell'asse delle ordinate ( asse Y)
                             ' per coordinate ad 1 o 2 quadranti (colore = bianco luminoso)

FOR x = .7 TO 4.201 STEP .001 ' campo di variabilità ed incremento della x (punto 1° e 2°)

PSET ( 109.5 * x , 160 - 111.491 * LOG ( x ) ) ,2 ' istruzione elaborata al punto 7°

NEXT x ' comanda il programma al ritorno automatico all'istruzione FOR x = .7 ecc.
```

F5

si ha la presentazione del grafico di Y = ln x, a tutto schermo, mostrata in figura 9.
Tracciata la curva ordinaria andiamo a valutare i valori degli intervalli in cui sono stati divisi gli assi cartesiani:
(asse X) ampiezza intervallo =
Valore massimo del campo / numero degli intervalli = 4.201 / 20 = .21
(asse Y) ampiezza intervallo =
Valore massimo del campo / numero degli intervalli = 1.435084/10 = .1435
visti i numeri si comprende che non è molto agevole valutare a vista sul grafico quale livello assume la y per un dato valore della x dato che gli intervalli in cui sono suddivisi i due assi cartesiani sono espressi da valori a più cifre.

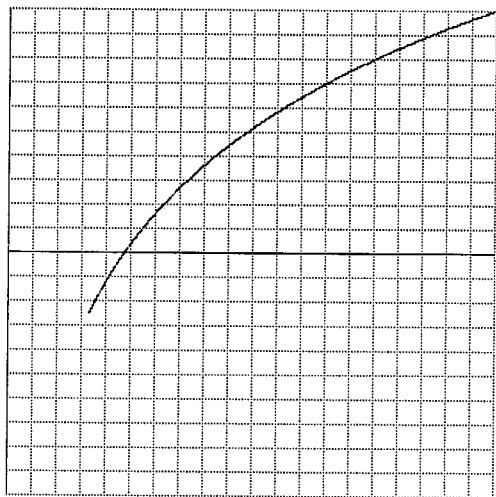



Figura 9
Grafico della funzione ln x
Campo di variabilità della x :
da .7 a 4.201
Scala asse x = .21 /div.
Scala asse y = .1143 / div.

Ecco quindi che se devono essere svolte sul grafico operazioni di lettura diretta nasce l'esigenza di calibrare gli intervalli del reticolo.
Per eseguire questa nuova operazione si deve rinunciare alla presentazione a tutto schermo del grafico della funzione per farla rientrare in limiti più ristretti; il procedimento è il seguente:

a) in base alla natura della funzione si sceglie la presentazione cartesiana a 2 quadranti "sbloccando" la prima e la quarta istruzione del programma di paragrafo 3.14
b) in base alla presentazione scelta si evidenziano il numero degli intervalli in cui sono divisi gli assi cartesiani: asse X = 20 intervalli asse Y = 10 intervalli
c) si riportano i valori del campo di variabilità della x: da .7 a 4.2
si sceglie un nuovo estremo del campo di variabilità di x, di poco superiore a 4.2, tale che sia un **intero divisibile per il numero degli intervalli**; nel nostro caso si può scegliere 5 ottenendo per gli intervalli il valore di 5 / 20 = .25
d) si riporta il valore massimo del campo di variabilità di y: 1.435084
si sceglie un numero, superiore a 1.435084, che sia un **intero divisibile per il numero degli intervalli**; nel nostro caso si sceglie il valore 2 ottenendo per gli intervalli il valore 2 / 10 = .2
e) si fissa l'incremento della variabile indipendente in .001
f) si riporta l'istruzione PSET relativa ai 2 quadranti **PSET (k1 * x , 160 - k2 * y)**
g) si determina il coefficiente k1 in base al nuovo campo di variabilità fissato per la x in 5
k1 = 460 / 5 = 92
h) si determina il coefficiente k2 in base al nuovo semicampo fissato per y in 2
k2 = 160 / 2 = 80
i) sulla base dei punti f), g), h), si completa, in simbologia Qbasic, l'istruzione finale che ci permette di tracciare la nostra funzione su scala calibrata, si aggiunge il colore giallo:

PSET (92 * x , 160 - 80 * LOG (x)) ,14

l) si procede alla compilazione del programma aggiungendo le istruzioni LOCATE che nei sistemi di assi cartesiani calibrati sono utili:

```
LINE ( 0 , 160 ) - ( 460 , 160 ) ,15  ' comanda il tracciamento dell'asse delle ascisse (asse X)
                                      ' per coordinate a 2 e a 4 quadranti (colore = bianco luminoso)

LINE ( 0 , 0 ) - ( 0 , 320 ) ,15  ' comanda il tracciamento dell'asse delle ordinate ( asse Y)
                                  ' per coordinate ad 1 o 2 quadranti (colore = bianco luminoso)

FOR x = .7 TO 5 STEP .001 ' campo di variabilità ed incremento della x , punto c) , punto e)

PSET ( 92 * x , 160 - 80 * LOG ( x ) ), 14 ' istruzione elaborata al punto i)

NEXT x ' comanda il programma al ritorno automatico all'istruzione FOR x = .7 ecc.

LOCATE 23 , 66 : PRINT "Y- Div. = . 2"   ' produce la scritta Y-Div. = . 2 nell'angolo
                                         ' basso a destra dello schermo

LOCATE 22 , 66 : PRINT "X- Div. = . 25"  ' produce la scritta X-Div: = . 25 nell'angolo
                                         ' basso a destra dello schermo
```

F5

si ha la presentazione del grafico di Y = ln x su assi calibrati ad intervalli di .25 per asse X e .2 per asse Y. come mostrato in figura 10.

Dall'esame di questo tracciato calibrato si osserva che è molto facile la lettura dei valori di y una volta assegnati quelli di x dato che per entrambi è possibile, contando il numero delle divisioni, risalire ai livelli cercati.

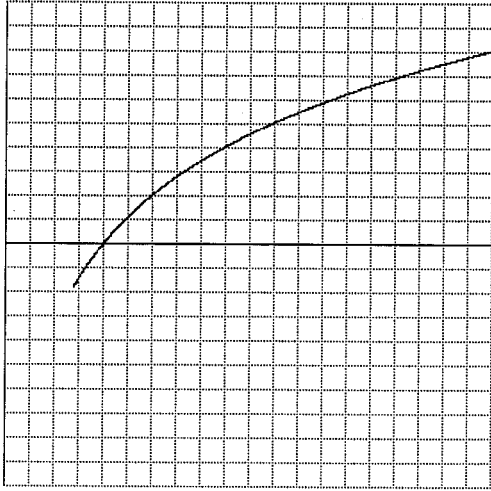



Figura 10
Grafico calibrato funzione ln x
Campo di variabilità della x :
da .7 a 5
Scala asse x = .25 /div.
Scala asse y = .2 / div.

3.28 Le funzioni a più valori

Le funzioni a più valori o polidrome hanno tre esempi nella geometria analitica: la circonferenza, l'ellisse e la parabola con asse orizzontale.
Queste funzioni, in particolare, presentano due valori della variabile dipendente in corrispondenza di un solo valore della variabile indipendente; si vedrà come si può affrontare semplicemente il problema della compilazione del programma per il tracciamento di tali curve.

3.28.1 Esercitazione grafica n° 17 (la circonferenza)

Per il tracciamento della circonferenza si devono impiegare le corrispondenze simboliche che riportiamo:

31) (circonferenza)

$Y = +/- (r^2 - x^2)^{1/2}$

Y = SQR (r ^ 2 - x ^ 2)
Y = - SQR (r ^ 2 - x ^ 2)

si tratta della forma più semplice di circonferenza, curva che ha il centro nell'origine degli assi cartesiani ed è definita dal solo raggio r.
Come anticipato al paragrafo 3.28 la funzione è a due valori, questi sono espressi dalle due simbologie in Qbasic che differiscono tra loro soltanto per i segni algebrici; ci troviamo perciò di fronte ad una nuova situazione che richiede un diverso approccio alla compilazione del programma per il tracciamento del grafico di una singola funzione.
Si deve pensare alla nuova curva come se fossero due distinte, da tracciare contemporaneamente

sullo schermo; una rappresenta il ramo di circonferenza di raggio r da tracciare nei due quadranti superiori del sistema di assi cartesiani, l'altra rappresenta il ramo di circonferenza di raggio r da tracciare nei due quadranti inferiori. Ciò premesso risulta evidente che si può far ricorso all'esperienza condotta nel paragrafo 3.20 in merito alla presentazione contemporanea di più funzioni e, fissato ad esempio il valore del raggio r = 5, scrivere:

-Per il ramo superiore (positivo) della circonferenza
1° dato il valore di r = 5 si stabilisce il campo di variabilità della x tra - 5 e + 5
2° si fissa l'incremento dei passi della variabile indipendente in .01
3° in base alla natura della funzione si sceglie la presentazione cartesiana a 4 quadranti "sbloccando" la prima e la terza istruzione del programma di paragrafo 3.14
4° si riporta l'istruzione PSET relativa ai 4 quadranti **PSET (230 + k * x , 160 - k2 * y)**
5° si determina il coefficiente k in base al semicampo di variabilità fissato per la x in 5
k = 230 / 5 = 46
6° si determina il coefficiente k2 in base al campo di variabilità previsto per y, che per la circonferenza è uguale ad r = 5
k2 = 160 / 5 = 32
7° sulla base dei punti 4°, 5°, 6°, si completa, in simbologia Qbasic, l'istruzione finale che ci consente di tracciare la nostra funzione, si aggiunge il colore verde:

PSET (230 + 46 * x , 160 - 32 * SQR (25 - x ^ 2)) ,2

-Per il ramo inferiore (negativo) della circonferenza l'impostazione è identica; la differenza è nell'istruzione finale che deve avere il segno negativo davanti alla funzione:

PSET (230 + 46 * x , 160 - 32 * (- SQR (25 - x ^ 2))) ,2

Con le due istruzioni che abbiamo costruito compiliamo il programma voluto:

```
LINE ( 0 , 160 ) - ( 460 , 160 ) ' comanda il tracciamento dell'asse delle ascisse (asse X) per 4 quadranti

LINE ( 230 , 0 ) - ( 230 , 320 ) ' comanda il tracciamento dell'asse delle ordinate ( asse Y) per 4 quadranti

FOR x= -5 TO 5 STEP .01 ' campo di variabilità ed incremento della x (punti 1° e 2°)

PSET ( 230 + 46 * x, 160 - 32 * SQR ( 25 - x ^ 2 ) ) ,2 ' gestisce il tracciamento del ramo di curva
                                                       ' superiore ( colore verde)

PSET ( 230 + 46 * x, 160 - 32 * ( - SQR ( 25 - x ^ 2 ) ) ) ,2 ' gestisce il tracciamento del ramo di
                                                              ' curva inferiore (colore verde)

NEXT x ' rimanda il programma al ritorno automatico all'istruzione FOR x = - 5 ecc.

F5
```

si ha la presentazione della circonferenza come mostrato in figura 11.

E' di notevole interesse didattico il tracciamento della circonferenza con il centro non disposto nell'origine degli assi cartesiani.
Numerosi problemi di geometria analitica impiegano questo tipo di curva.
Lasciamo al lettore, a titolo di esercizio, la composizione dell'uguaglianza simbolica per tale circonferenza è la sua implementazione in Qbasic.

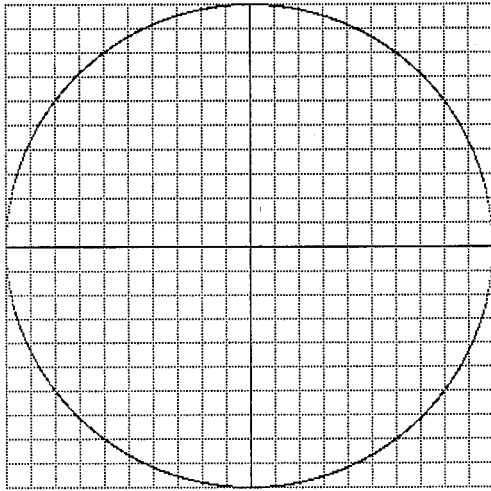



Figura 11
Grafico funzione circonferenza
Campo di variabilità della x :
da -r a +r
Scala asse x = .5 /div.
Scala asse y = .5/ div.

3.28.2 Esercitazione grafica n 18° (l'ellisse)

Così come abbiamo fatto nell'esercitazione precedente riportiamo le corrispondenze simboliche che interessano l'ellisse:

33) (ellisse)

$Y = +/- (b : a) (a^2 - x^2)^{1/2}$

Y = (b / a) * SQR (a ^ 2 - x ^ 2)
Y = - (b / a) * SQR (a ^ 2 - x ^ 2)

Anche in questo caso, come per la circonferenza, la curva deve essere divisa idealmente in due rami e, una volta fissati i parametri a; b, si procede alla compilazione del programma con un metodo un poco diverso dal precedente per dar modo al lettore di apprendere un nuovo sistema di impostazione. Assumendo b = 3, a = 6 scriviamo:
-Per il ramo superiore (positivo) dell'ellisse
1° dato il valore di a = 6 si stabilisce il campo di variabilità della x tra - 6 e + 6
2° si fissa l'incremento dei passi della variabile indipendente in .01
3° in base alla natura della funzione si sceglie la presentazione cartesiana a 4 quadranti "sbloccando" la prima e la terza istruzione del programma di paragrafo 3.14
4° si riporta l'istruzione PSET relativa ai 4 quadranti **PSET (230 + k * x , 160 - k2 * y)**
5° si determina il coefficiente k in base al semicampo di variabilità fissato per la x in 6
k = 230 / 6 = 38.33
6° per non deformare il tracciato dell'ellisse si calcola k2 in modo da ottenere per l'asse Y la stessa scala assegnata all'asse X k2 = 160 / 6 = 26.66
7° sulla base dei punti 4°, 5°, 6°, si completa, in simbologia Qbasic, la funzione per il ramo positivo

Y = (3 / 6) * (SQR (36 - x ^ 2))

8° si costruisce l'istruzione di PSET con il valore di Y sopra impostato (curva di colore giallo)

PSET (230 + 38.33 * x , 160 - 26.66 * Y) ,14

-Per il ramo inferiore (negativo) dell'ellisse basta scrivere la nuova istruzione ponendo t = -Y

PSET (230 + 38.33 * x , 160 - 26.66 * t) ,14

con le due istruzioni che abbiamo costruito compiliamo il programma voluto:

```
LINE ( 0 , 160 ) - ( 460 , 160 )  ' comanda il tracciamento dell'asse delle ascisse (asse X)
                                   ' per coordinate a 2 e a 4 quadranti (colore = bianco luminoso)

LINE ( 230 , 0 ) - ( 230 , 320 )  ' comanda il tracciamento dell'asse delle ordinate ( asse Y)
                                   ' per coordinate a 4 quadranti  (colore = bianco luminoso)

FOR  x= - 6  TO  6  STEP .01  ' campo di variabilità ed incremento della x (punti 1° e 2°)

Y = ( 3 / 6 ) * ( SQR ( 36 - x ^ 2 ) )  ' calcolo della funzione del ramo positivo dell'ellisse

t = - Y  ' cambiamento di segno della funzione per il calcolo del ramo negativo dell'ellisse

PSET ( 230 + 38.33 * x , 160 - 26.66 * Y ) ,14  ' gestisce il tracciamento del ramo di
                                                 ' curva positiva ( colore giallo)

PSET ( 230 + 38.33 * x , 160 - 26.66 * t ),14  ' gestisce il tracciamento del ramo di
                                                ' curva  negativa (colore giallo)

NEXT x  ' rimanda il programma al ritorno automatico all'istruzione FOR x = - 6 ecc.
```

F5
si ha la presentazione dell'ellisse come mostrato in figura 12.

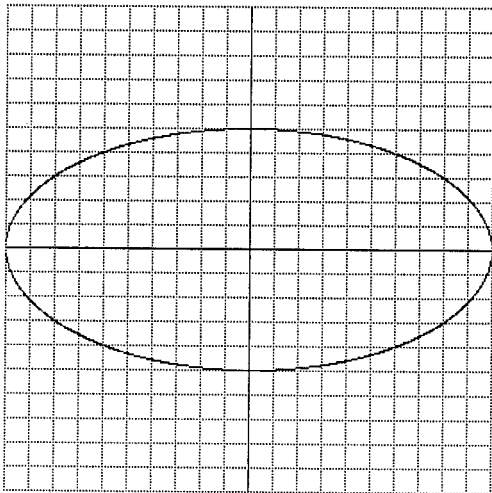



Figura 12
Grafico della funzione ellisse
Campo di variabilità della x :
da -6 a + 6
Scala asse x = .6 /div.
Scala asse y = .6 / div.

3.28.3 Esercitazione grafica n° 19 (la parabola con asse orizzontale)

Eseguiamo quest'ultimo esercizio sulle funzioni polidrome compilando il programma per il tracciamento del grafico della parabola ad asse orizzontale in base alle corrispondenze simboliche:

34) (parabola con asse orizzontale)

$Y = +/- (2 p x)^{1/2}$

Y = SQR (2 * p * x)
Y = - SQR (2 * p * x)

procediamo secondo lo schema dell'esercitazione n° 18 dopo aver assunto per p il valore + 3:

-Per il ramo superiore (positivo) della parabola
1° dato il valore positivo di p si stabilisce il campo di variabilità della x tra 0 e + 12
2° si fissa l'incremento dei passi della variabile indipendente in .01
3° in base alla natura della funzione si sceglie la presentazione cartesiana a 2 quadranti "sbloccando" la prima e la quarta istruzione del programma di paragrafo 3.14
4° si riporta l'istruzione PSET relativa ai 2 quadranti **PSET (k1 * x , 160 - k2 * y)**
5° si determina il coefficiente k1 in base al semicampo di variabilità fissato per la x in 12
k1 = 460 / 12 = 38.33
6° si determina il coefficiente k2 in base al campo di variabilità previsto per y, che per la parabola in oggetto è uguale al valore che la funzione assume per x = 12; Y = 8.48
k2 = 160 / 8.48 = 18.86
7° sulla base dei punti 4°, 5°, 6°, si completa, in simbologia Qbasic, la funzione per il ramo positivo

Y = SQR (2 * 3 * x)

8° si costruisce l'istruzione di PSET con il valore di Y sopra impostato (curva di colore verde)

PSET (38.33 * x , 160 - 18.86 * Y),2

-Per il ramo inferiore (negativo) della parabola basta scrivere la nuova istruzione ponendo t = -Y

PSET (38.33 * x , 160 - 18.86 * t),2

con le due istruzioni che abbiamo costruito compiliamo il programma voluto:

```
LINE ( 0 , 160 ) - ( 460 , 160 ) ' comanda il tracciamento dell'asse delle ascisse (asse X) a 2 quadranti

LINE ( 0 , 0 ) - ( 0 , 320 )    ' comanda il tracciamento dell'asse delle ordinate ( asse Y) a 2 quadranti

FOR x = 0 TO 12 STEP .01 ' campo di variabilità ed incremento della x (punti 1° e 2°)

Y = SQR ( 2 * 3 * x ) ' calcolo della funzione del ramo positivo della parabola

t = - Y ' cambiamento di segno della funzione per il calcolo del ramo negativo dell'ellisse

PSET ( 38.33 * x , 160 - 18.86 * Y ) ,2   ' gestisce il tracciamento del ramo di curva positiva ( colore verde)

PSET ( 38.33 * x , 160 - 18.86 * t ) ,2   ' gestisce il tracciamento del ramo di curva negativa (colore verde)

NEXT x ' rimanda il programma al ritorno automatico all'istruzione FOR x = 0 ecc.
```

F5

si ha la presentazione della parabola con asse orizzontale come mostrato in figura 13.

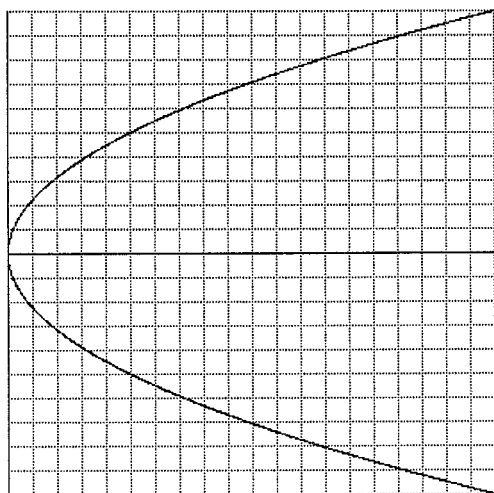



Figura 13
Grafico funzione parabola
Campo di variabilità della x :
da 0 a 12
Scala asse x = .6 /div.
Scala asse y = .6 / div.

3.29 Le funzioni di tabella

Concludiamo il presente capitolo con una interessante applicazione del metodo grafico Qbasic che permette di tracciare gli andamenti delle "funzioni di tabella". Le funzioni di tabella sono formate da un insieme discreto di valori numerici associati in corrispondenza con altri valori; un esempio aiuta a chiarire il concetto:

Siano state rilevate ad intervalli regolari di un' ora quattro letture relative al consumo di acqua in un appartamento, con i valori rilevati sia stato compilato il seguente prospetto:

Tempo in ore	Consumo in metri cubi
1	.2
2	.3
3	.7
4	.8

è ovvio che il consumo dell'acqua dipende, con legge casuale, dal tempo che passa; il consumo è la variabile dipendente ed il tempo la variabile indipendente.
Il prospetto sopra indicato è di fatto una tabella in cui nella prima colonna è riportata la variabile indipendente (x) e nella seconda colonna la variabile dipendente (y). Tabelle simili, contenenti molte coppie di valori possono essere il frutto di attività varie, sia dovute a sviluppi di carattere matematico, sia dovute a rilievi sperimentali di fenomeni fisici.
Spiegata la dizione "funzioni di tabella" si comprende come questo particolare tipo di funzione

possa richiedere il tracciamento grafico del suo andamento, sia per evidenziarne particolari caratteristiche, sia per favorirne il controllo comparativo con funzioni matematiche precalcolabili od altro.
L'insieme delle coppie di valori che formano una tabella prende il nome di " **matrice**".
-La matrice può essere di "tipo permanente" e come tale memorizzata nel contesto di un programma; in tal caso essa resta in archivio con il programma stesso per essere impiegata ogni qualvolta sia necessario. La matrice permanente richiede un sensibile tempo di inserimento nel programma .
-La matrice può essere di "tipo volatile" e come tale impiegabile in un programma dopo averla inserita, non può essere memorizzata con il programma ma deve essere reinserita ogni volta che se ne richiede l'utilizzazione. La matrice volatile è inseribile nel contesto del programma con rapidità.
In questo capitolo tratteremo soltanto delle matrici di tipo permanente rimandando il lettore al capitolo 7 per la descrizione delle matrici di tipo volatile.
Per tracciare il grafico di una matrice permanente è richiesta la compilazione di un programma che impiega alcune nuove istruzioni; ne diamo spiegazioni nell'ordine in cui intervengono, intercalate ad altre già note, nell'ambito della routine:

DIM a(n) questa istruzione fissa il numero n delle coppie che formano la matrice; se le coppie sono ad esempio 65 si dovrà scrivere **DIM a(65)**.
GOSUB matrice questa istruzione porta l'esecuzione del programma nella zona dello stesso dove è stata compilata la matrice.
END questa istruzione impone al programma di fermarsi dopo che sono stati letti tutti i valori che compongono la matrice.
matrice: è il nome assegnato alla matrice e richiamato dall'istruzione GOSUB; il nome può essere qualsiasi, in questa esposizione è stato scelto " matrice".
a(x) = y è l'istruzione che identifica una generica coppia dei valori di matrice; in cui **x** è il valore della variabile indipendente e **y** è il valore della variabile dipendente; ad esempio per la coppia x = 37 e Y = -1.56 scriveremo: **a(37) = -1.56.**
RETURN è l'istruzione che consente di ritornare al punto di programma dal quale si esplora la matrice; questa istruzione deve essere collocata alla fine della matrice.

Con le indicazioni che abbiamo dato rimandiamo il lettore all'esercitazione grafica n° 20 nella quale applichiamo questo nuovo gruppo di istruzioni per la compilazione di un istruttivo programma di grafica.

3.30 Esercitazione grafica n° 20 (funzione di tabella)

Proponiamoci di tracciare il grafico della funzione di tabella sotto riportata:

variabile indipen. x	variabile dipen. y
0	.2
1	.5
2	.8
3	1.2
4	1.7
5	2.3
6	3.6
7	4.6
8	5.2
9	6

La funzione di tabella è formata con 34 coppie di valori che devono essere rappresentati mediante un grafico a punti discreti; ciascun punto deve comparire sullo schermo video in base alle coordinate x , y definite dalla tabella stessa . La settima coppia di valori , ad esempio, deve definire la posizione del settimo punto del grafico in base alle coordinate x = 6 y = 3.6; la trentesima coppia di valori deve definire la posizione del trentesimo punto in base alle coordinate

10	6.5
11	7.2
12	7.8
13	8.1
14	8.2
15	8.2
16	7.8
17	7.2
18	6.5
19	6.1
20	6
21	6
22	6.1
23	6.2
24	6.3
25	6.1
26	5.8
27	4.8
28	3.8
29	2.8
30	2
31	1.5
32	1.2
33	1

x = 29 y = 2.8 .

Si deve osservare che le coppie , pur essendo 34, hanno come valore estremo di x il numero 33, ciò per il fatto che il primo valore di x è uguale a 0.

I valori che formano la funzione di tabella saranno chiamati nel prosieguo, indifferentemente, tabella o matrice.

Iniziamo ad elaborare i primi dati per la compilazione del programma grafico:

1° si riporta il campo di variabilità della x dato dalla matrice: da 0 a 33
2° l'incremento dei passi della variabile indipendente è stabilito dalla matrice ad 1
3° in base alla natura della funzione (che la matrice mostra sempre positiva) si sceglie la presentazione ad 1 quadrante "sbloccando" la seconda e la quarta istruzione del programma di paragrafo 3.14
4° si riporta l'istruzione PSET relativa ad 1 quadrante **PSET (k1 * x , 320 - k3 * y)**
5° si determina il coefficiente k1 in base al campo di variabilità fissato per la x in 33
k1 = 460 / 33 = 13.93
6° si determina il coefficiente k3 in base all'estremo del campo di variabilità di y che la matrice fissa in 8.2
K3 = 320 / 8.2 = 39.02
7° sulla base dei punti 4°, 5°, 6°, si completa l' istruzione PSET dove y = a(x):

PSET (13.93 * x , 320 - 39.02 * a(x))

8° sulla base delle dimensioni della matrice si fissa l'istruzione DIM; DIM a (34)
si procede ora alla compilazione del programma:

LINE (0 , 320) - (460 , 320) ' comanda il tracciamento dell'asse delle ascisse (asse X) coordinate I quadrante

LINE (0 , 0) - (0 , 320) ' comanda il tracciamento dell'asse delle ordinate (asse Y) coordinate I quadrante

DIM a (34) ' fissa la dimensione della matrice da utilizzare

FOR x = 0 TO 33 ' impone la scansione dei valori della matrice l'uno dopo l'altro

GOSUB matrice ' invia alla matrice per la lettura dei valori di a(x)

PSET (13.93 * x , 320 - 39.02 * a(x)) ' comanda la presentazione dei punti che rappresentano la matrice &

```
PSET ( 13.93 * x + 1 , 320 - 39.02 * a( x ) ) ' seguono quattro istruzioni per evidenziare i punti

PSET ( 13.93 * x - 1 , 320 - 39.02 * a( x ) ) ' mediante crocette ( tracce più visibili dei singoli punti)

PSET ( 13.93 * x , 320 - 39.02 * a( x ) + 1 )

PSET ( 13.93 * x , 320 - 39.02 * a( x ) - 1 )

NEXT x   ' comanda il ritorno all'istruzione FOR

END  ' comanda la fermata del programma una volta che tutti i valori della matrice sono stati letti


matrice :     ' nome della matrice da esplorare su comando GOSUB

a(0) = .2    ' dal termine a( 0 ) al termine a( 33 ) sono raccolti i 34 valori di y trascritti dalla tabella
a(1) = .5    ' originale data all'inizio dell'esercizio
a(2) = .8
a(3) = 1.2
a(4) = 1.7
a(5) = 2.3
a(6) = 3.6
a(7) = 4.6
a(8) = 5.2
a(9) = 6
a(10) = 6.5
a(11) = 7.2
a(12) = 7.8
a(13) = 8.1
a(14) = 8.2
a(15) = 8.2
a(16) = 7.8
a(17) = 7.2
a(18) = 6.5
a(19) = 6.1
a(20) = 6
a(21) = 6
a(22) = 6.1
a(23) = 6.2
a(24) = 6.3
a(25) = 6.1
a(26) = 5.8
a(27) = 4.8
a(28) = 3.8
a(29) = 2.8
a(30) = 2
a(31) = 1.5
a(32) = 1.2
a(33) = 1

RETURN ' istruzione per il ritorno al punto di programma dove si comanda la lettura della matrice
```

F5

si ha la presentazione della punteggiata che mostra l'andamento della funzione di tabella come illustrato in figura 14.

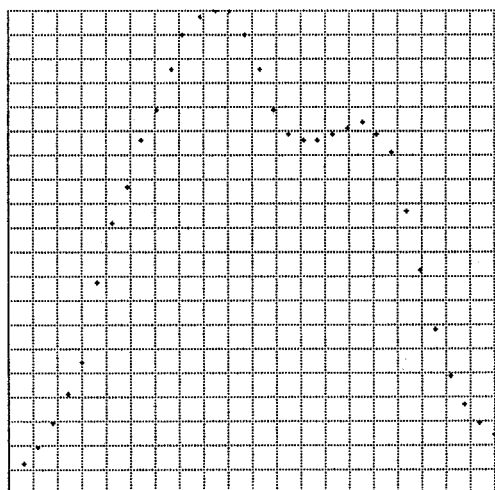



Figura 14
Grafico funzione di tabella
Campo di variabilità della x :
da 0 a 33
Scala asse x = .6 / div.
Scala asse y = .41 / div.

CAPITOLO 4
LA SOLUZIONE DELLE EQUAZIONI ALGEBRICHE E TRASCENDENTI

In questo capitolo inizieremo a mettere in pratica ciò che abbiamo imparato in precedenza allo scopo di acquisire strumenti di calcolo che ci diano modo di risolvere rapidamente diverse tipologie di equazioni e di sistemi di equazioni di tipo algebrico e trascendente .
Procederemo allo scopo per successivi gradi di difficoltà, alternando informazioni di carattere teorico ad esercizi di programmazione mirati alla soluzione dei nostri problemi; il lettore non si dovrà stupire se alcuni esercizi saranno molto semplici, essi costituiranno i mattoni con i quali costruire strutture più complesse.
La raccolta dei programmi che compileremo costituirà un piccolo archivio nel quale il lettore potrà ricercare, quando ne avrà la necessità, quanto di aiuto al proprio lavoro.

4.1 L' equazione algebrica di primo grado

Come è noto l'equazione algebrica di primo grado si presenta nella forma:

$$a x + b = 0$$

dalla quale si ricava con immediatezza la soluzione

$$x = - b / a$$

Per risolvere questo tipo di equazione mediante il Qbasic, anche se ciò è ovviamente superfluo, si procede alla compilazione del programma che riportiamo e commentiamo:

```
CLS ' pulizia dello schermo

INPUT " a = " ; a  ' richiesta di ingresso del coefficiente ( a ) dell'incognita x

INPUT " b = " ; b  ' richiesta di ingresso del termine noto ( b )

x= -( b / a )  ' calcolo del valore dell'incognita

PRINT " x = " ; x  ' presentazione del valore che risolve l'equazione
```

Per risolvere ad esempio l'equazione $3x - 7 = 0$ si esegue il programma inserendo i dati richiesti

```
                F5
a = ? 3
b= ? -7
x = 2.333333
                F5
```

4.1.1 La soluzione grafica dell'equazione di primo grado

L'equazione di primo grado, risolta algebricamente nel paragrafo 4.1, è risolvibile con un certo grado di approssimazione anche per via grafica; l'esercizio richiesto per questa operazione, esposto a solo scopo didattico, è la base per significativi sviluppi che saranno oggetto del prosieguo del nostro studio.

Per affrontare il problema della soluzione grafica dell'equazione citata è necessario considerare detta equazione come un caso particolare della funzione di geometria analitica riguardante la retta; infatti tale funzione è esplicitata mediante l'espressione:

$$Y = m x + n$$

che per $Y = 0$ coincide con l'equazione di primo grado in x; in altre parole la soluzione dell'equazione di primo grado altro non è che il valore della variabile indipendente x che azzera la variabile dipendente Y, è cioè il punto in cui la retta interseca l'asse delle ascisse .
Se consideriamo ad esempio l'equazione $4 x + 8 = 0$ questa è il caso particolare della funzione

$$Y = 4 x + 8$$

in cui $m = +4$ ed $n = +8$.

Se tracciamo perciò il grafico della retta $Y = 4 x + 8$ il valore dell'ascissa in cui la retta stessa incontra l'asse x risolve l'equazione data.
Vediamo come procedere per ottenere questo risultato:
E' necessario anzitutto stabilire l'impiego di un sistema di assi cartesiani calibrato e pertanto seguire la procedura indicata nel paragrafo 3.27:
a) in base alla natura della funzione si sceglie la presentazione cartesiana a 4 quadranti "sbloccando" la prima e la terza istruzione del programma di paragrafo 3.14
b) in base alla presentazione scelta si evidenziano il numero degli intervalli in cui sono divisi gli assi cartesiani: asse X = 10 intervalli asse Y = 10 intervalli
c) si fissano i valori del campo di variabilità della x in modo tale che la y corrispondente compia certamente una escursione dai valori negativi a quelli positivi: ciò si ottiene, a vista, per x che varia da -10 a +10, con questi valori si ha un adattamento automatico al numero degli intervalli stabiliti per la calibrazione della scala
d) si riporta il valore massimo del campo di variabilità di Y: 48, si sceglie un numero, superiore a 48,che sia un **intero divisibile per il numero degli intervalli**; nel nostro caso si sceglie il valore 50 ottenendo per gli intervalli il valore 50 / 10 = 5
e) si fissa l'incremento della variabile indipendente in .01
f) si riporta l'istruzione PSET relativa ai 4 quadranti **PSET (230 + k * x , 160 - k2 * y)**
g) si determina il coefficiente k in base al campo di variabilità fissato per la x in 10
k = 230 / 10 = 23
h) si determina il coefficiente k2 in base al nuovo semicampo fissato per y in 50
k2 = 160 / 50 = 3.2
i) sulla base dei punti f), g), h), si completa, in simbologia Qbasic, l'istruzione finale che ci consente di tracciare la nostra retta su scala calibrata, si aggiunge il colore giallo:

PSET (230 + 23 * x , 160 - 3.2 * (4 * x + 8)) ,14

l) si procede alla compilazione del programma aggiungendo le istruzioni LOCATE

```
LINE ( 0 , 160 ) - ( 460 , 160 )  ' comanda il tracciamento dell'asse delle ascisse (asse X)
                                  ' per coordinate a 2 e a 4 quadranti (colore = bianco luminoso)

LINE ( 230 , 0 ) - (230 , 320 )  ' comanda il tracciamento dell'asse delle ordinate ( asse Y)
                                 ' per coordinate a 4 quadranti (colore = bianco luminoso)

FOR x = - 10 TO 10 STEP .01  ' campo di variabilità ed incremento della x , punto c) , punto e)      &
```

```
PSET ( 230 + 23 * x , 160 - 3.2 * ( 4 * x + 8 ) ) ,14   ' istruzione elaborata al punto  i)

NEXT x  ' comanda il programma al ritorno automatico  all'istruzione FOR x =  -10  ecc.

LOCATE 23 , 66 : PRINT "Y- Div. = 5"     ' produce la scritta     Y-Div. = 5 nell'angolo
                                         ' basso a destra dello schermo

LOCATE 22 , 66 : PRINT "X- Div. = 1"     ' produce la scritta     X-Div: =  1   nell'angolo
                                         ' basso a destra dello schermo
```

F5

si ottiene il grafico riportato in figura 15.

Dal grafico di figura 15 si osserva che il punto di intersezione tra la retta, di equazione $Y = 4x + 8$ e l'asse delle ascisse si ha per $x = -2$, valore questo che risolve l'equazione $4x + 8 = 0$ che ci eravamo proposta come esempio.
E' evidente che un procedimento così elaborato, come quello eseguito per ottenere la soluzione dell'equazione data, non è giustificabile se non per mostrare, in modo semplice, un tipo di metodologia che sarà invece indispensabile, in seguito, per la ricerca delle soluzioni di particolari equazioni.

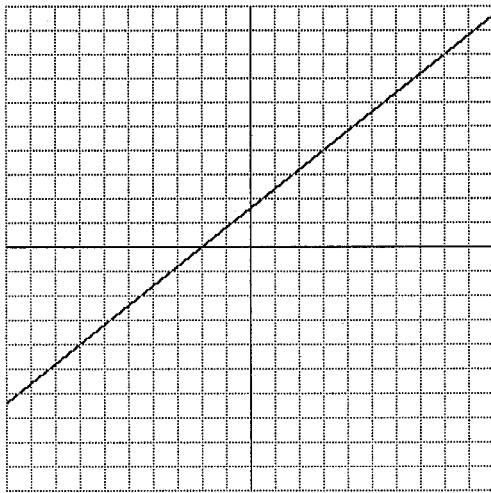



Figura 15
Grafico per soluzione equazione algebrica di primo grado
Campo di variabilità della x :
da -10 a +10
Scala asse x = 1 /div.
Scala asse y = 5 / div.

4.2 I sistemi di equazioni algebriche di primo grado a due incognite

In molte applicazioni matematiche e tecniche capita sovente di dover risolvere un sistema di due equazioni del tipo:

$$\begin{cases} a1\ x + b1\ y = c1 \\ a2\ x + b2\ y = c2 \end{cases}$$

Dal punto di vista operativo la soluzione del sistema non presenta difficoltà, se però si devono risolvere più sistemi diversi, con coefficienti numerici a più cifre l'operazione diventa pesante e richiede molta attenzione per non incorrere in errori.

Questo è pertanto il caso in cui l'impiego del P.C. in Qbasic sveltisce notevolmente il lavoro necessario per la soluzione dei sistemi di equazioni.
Il programma che andiamo a compilare si basa sulle due forme risolutive del sistema espresse in base ai coefficienti che vi compaiono:

$$x = \frac{c1\ b2 - b1\ c2}{a1\ b2 - b1\ a2}$$

$$y = \frac{a1\ c2 - c1\ a2}{a1\ b2 - b1\ a2}$$

formule che opportunamente tradotte in linguaggio Qbasic rendono la seguente stesura:

```
CLS ' pulisce lo schermo

INPUT " a1= " ; a1  ' richiesta di ingresso coefficiente a1

INPUT " b1= " ; b1  ' richiesta di ingresso coefficiente b1

INPUT " c1= " ; c1  ' richiesta di ingresso coefficiente c1

INPUT " a2= " ; a2  ' richiesta di ingresso coefficiente a2

INPUT " b2= " ; b2  ' richiesta di ingresso coefficiente b2

INPUT " c2= " ; c2  ' richiesta di ingresso coefficiente c2

x = ( c1 * b2 - b1 * c2 ) / ( a1 * b2 - b1 * a2 )  ' calcolo dell'incognita x

y = ( a1 * c2 - c1 * a2 ) / ( a1 * b2 - b1 * a2 )  ' calcolo dell'incognita y

PRINT " x= " ; x  ' visualizzazione soluzione in x

PRINT " y=" ; y  ' visualizzazione soluzione in y
```

Per renderci conto della comodità del programma lo proviamo per la soluzione del seguente sistema di due equazioni:

$$\begin{cases} -3.4321\ x - 579.56\ y = 344.78 \\ 45.897\ x - 345.99\ y = -217 \end{cases}$$

sistema che, come si può vedere dai valori dei coefficienti, non è facilmente manipolabile.
Facciamo girare il programma:

```
              F5
a1= ?    -3. 4321
b1= ?  -579. 56
c1= ?   344. 78
a2 = ?    45. 897
b2 = ? - 345. 99
c2 = ? - 217
x =  - 8.81888
y =  - . 542675
              F5
```

Il sistema che abbiamo risolto, con l'ausilio del programma testé compilato, mostra l'utilità di questa rapida procedura che ci dà la possibilità di affrontare, in brevissimo tempo, la soluzione di sistemi algebrici che richiederebbero altrimenti tempi di elaborazione non indifferenti.
E' importante osservare che un sistema ammette soluzioni quando è **determinato**, se invece è indeterminato od impossibile non si hanno soluzioni e il programma si blocca; compare in tale caso un riquadro, al centro dello schermo, con la scritta **Overflow**. Dopo l'overflow, per riprendere il lavoro, è necessario pigiare il tasto **ESC** e riavviare il programma.

4.2.1 La soluzione grafica per i sistemi algebrici a due incognite

La soluzione dei sistemi algebrici per via grafica non ha interesse pratico, la procedura da seguire è però importante, sia come esercizio speculativo, sia come fondamentale preparazione per altre applicazioni nel campo delle equazioni trascendenti; vediamo quali sono i criteri che la informano:

Dal sistema

$$\begin{cases} a1\ x + b1\ y = c1 \\ a2\ x + b2\ y = c2 \end{cases}$$

esplicitando l'incognita y dalle due equazioni otteniamo una diversa struttura equivalente:

$$\begin{cases} y = -(a1 / b1)\ x + (c1 / b1) \\ y = -(a2 / b2)\ x + (c2 / b2) \end{cases}$$

dalla nuova struttura si vede che le due equazioni non sono altro che le funzioni della geometria analitica rappresentanti due rette; la soluzione del sistema infatti è data dai valori delle coordinate x ed y del punto di intersezione di queste rette, coordinate che si possono ricavare, con approssimazione, dal grafico delle due funzioni. Un esempio è necessario per entrare nel merito di questo argomento: proponiamoci, ad esempio, la soluzione grafica del sistema

$$\begin{cases} 3x + 5y = 7 \\ 8x - 3y = 12 \end{cases}$$

esplicitando le y delle due equazioni otteniamo la struttura equivalente del sistema

$$\begin{cases} y = (-3/5)\ x + 7/5 \\ y = (8/3)\ x - 4 \end{cases}$$

con la quale abbiamo messo in evidenza le equazioni delle due rette che dobbiamo tracciare.
Dobbiamo ora compilare il programma grafico che ci dia modo di valutare i valori delle coordinate x ed y del punto di intersezione delle rette; tali valori sono le soluzioni del sistema dato.
Definiamo pertanto:

a) in base alla natura del problema, la ricerca del punto di intersezione tra due rette che può essere ovunque collocato, si deve adottare la presentazione cartesiana a 4 quadranti "sbloccando" la prima e la terza istruzione del programma di paragrafo 3.14
b) in base alla presentazione scelta si evidenziano il numero degli intervalli in cui sono divisi gli assi cartesiani: asse X = 10 intervalli asse Y = 10 intervalli
c) si fissano i valori del campo di variabilità della x da -10 a +10; con questi valori si ha un adattamento automatico al numero degli intervalli stabiliti per la calibrazione della scala

d) si fissa il valore massimo del campo di variabilità di y per avere un adattamento alla scala y =10
e) si fissa l'incremento della variabile indipendente in .01 (pari a 1000 punti di calcolo)
f) si riporta l'istruzione PSET relativa ai 4 quadranti **PSET (230 + k * x , 160 - k2 * y)**
g) si determina il coefficiente k in base al campo di variabilità fissato per la x in 10
k = 230 / 10 = 23
h) si determina il coefficiente k2 in base al semicampo fissato per y in 10
k2 = 160 / 10 = 16
i) si riporta l' equazione della prima retta distinguendo la variabile dipendente y con y1

y1 = (-3 / 5) x + 7/5

l) si riporta l' equazione della seconda retta distinguendo la variabile dipendente y con y2

y2 = (8 / 3) x - 4

m) sulla base dei punti f), g), h), i), si completano, in simbologia Qbasic, le istruzioni finali che permettono di tracciare la prima retta su scala calibrata:

y1 = (-3 / 5) * x + (7 / 5)

PSET (230 + x * 23, 160 - 16 * y1)

n) sulla base dei punti f), g), h), l), si completano, in simbologia Qbasic, le istruzioni finali che consentono di tracciare la seconda retta su scala calibrata:

y2 = (8 / 3) * x - 4

PSET (230 + x * 23, 160 - 16 * y2)

si compila infine il programma:

```
LINE ( 230, 0 )-( 230, 320 )  ' ASSE  Y  a  4  quadranti

LINE ( 0, 160 )-( 460, 160 )  ' ASSE  X  a  4  quadranti

FOR  x = - 10  TO  10 STEP  .01  ' campo di variabilità imposto ed incremento della x

y1 = (  -3 / 5  ) * x  +  (  7 / 5  )   '  calcolo funzione  della prima retta

PSET (  230  +  x  *  23 , 160 - 16  *  y1 )  ' presentazione grafica prima retta

y2 = (  8 / 3  )  *  x  -  4   ' calcolo funzione della seconda retta

PSET (  230  +  x  *  23 , 160 - 16  *  y2  ) ' presentazione grafica seconda retta

NEXT x   ' rimanda all'istruzione FOR x= -10 ecc.

LOCATE 23, 66: PRINT "y-Div.= 1"    ' intervalli scala asse Y  ;  1

LOCATE 22, 66: PRINT "x-Div.= 1"    ' intervalli scala asse  X  ; 1

F5
```

si ha la presentazione delle due rette come mostrato in figura 16

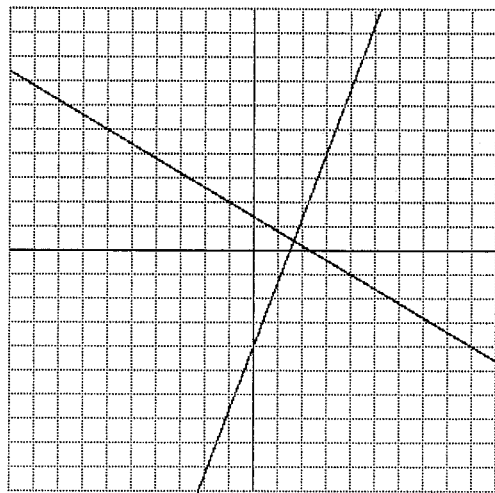



Figura 16
Grafico per la soluzione di sistema algebrico di primo grado a due incognite
Campo di variabilità della x : da -10 a +10
Scala asse x = 1 /div.
Scala asse y = 1 / div.

Dall'esame del grafico si vede che le rette si intersecano in un punto le cui coordinate x ed y possono essere valutate, con approssimazione, in: x = 1.6 y = .4; questi valori sono, in prima battuta, la soluzione del sistema dato. E' possibile, una volta individuato il punto di intersezione delle due rette, aumentare la precisione del rilievo grafico mediante opportuni cambiamenti di scala; vedremo più avanti, in più importanti applicazioni, come procedere per ottenere risultati migliori. Nel grafico una retta esce naturalmente dal reticolo; impareremo a troncarne il tracciato per contenere le immagini nell'ambito dello spazio assegnato.

4.3 L'equazione algebrica di secondo grado

In questo paragrafo vogliamo mostrare come si implementa in Qbasic un programma per la soluzione delle equazioni algebriche di secondo grado; anche questo strumento, così come quello relativo alla soluzione dei sistemi a due incognite, potrà essere archiviato per eventuali impieghi di lavoro.
L'equazione algebrica di secondo grado si presenta nella forma canonica:

$$a x^2 + b x + c = 0$$

in cui (a) e (b) rappresentano rispettivamente i coefficienti dell'incognita di secondo e di primo grado e (c) è il termine noto dell'equazione.
Le soluzioni, o radici, dell'equazione sono date dalle espressioni:

$$x1 = \frac{-b + (D)^{1/2}}{2a}$$

$$x2 = \frac{-b-(D)^{1/2}}{2a}$$

in cui $D = b^2 - 4ac$
rappresenta il "discriminante dell'equazione"

Per implementare le formule risolutive in Qbasic si deve ricordare:

- se il valore di D " discriminante dell'equazione" è maggiore od uguale a zero si hanno soluzioni reali

- se D è inferiore a zero si hanno soluzioni complesse coniugate.

Ciò impone al programma una valutazione del segno del discriminante in modo che la routine di calcolo possa stabilire quando le soluzioni dell'equazione sono reali o quando invece sono complesse.
Per questo tipo di elaborazione è utile trasformare le due espressioni risolutive come segue:

$$x1 = \frac{-b}{2a} + \frac{(D)^{1/2}}{2a}$$

$$x2 = \frac{-b}{2a} - \frac{(D)^{1/2}}{2a}$$

dalle quali indicando con E ed F le frazioni

$$E = \frac{-b}{2a} \qquad F = \frac{(D)^{1/2}}{2a}$$

si possono riscrivere le formule risolutive in modo da implementarle più facilmente nella routine di calcolo

$$x1 = E + F \qquad x2 = E - F$$

ciò fatto non resta che compilare il programma per la soluzione dell'equazione di secondo grado:

```
CLS ' pulisce lo schermo

INPUT " a = " ; a ' richiesta di ingresso per il coefficiente a

INPUT " b = " ; b ' richiesta di ingresso per il coefficiente b

INPUT " c = " ; c ' richiesta di ingresso per il termine noto c

D = b ^ 2 - 4 * a * c ' calcolo del valore del discriminante D

E = - ( b / ( 2 * a ) ) ' calcolo del valore di E

F = ( 1 / ( 2 * a ) ) * SQR ( ABS ( D ) ) ' calcolo del valore di F mediante il valore assoluto | D |; ciò &
```

```
                                  ' allo scopo di computare la radice quadrata anche se  D < 0
x1 =  E + F  ' calcolo della prima soluzione

x2 =  E - F  ' calcolo della seconda soluzione

IF  D >= 0  THEN  PRINT "x1=" ; x1 ; "                  " ; " x2= " ; x2
                  ' istruzione che comanda la 'stampa di x1 e x2  se sono valori reali ,cioè per  D>=0

IF  D < 0   THEN  PRINT "x1=" ; E ; " + j " ; ABS ( F ) ; "        " ; " x2 = " ; E ; "-j" ; ABS ( F )
                  ' istruzione che comanda la stampa di x1 e x2 se sono valori complessi
                  ' cioè per D < 0 ,  in tale caso stampa :
                  ' per x1 ;  + j davanti al valore assoluto di  F
                  ' per x2 ;  - j davanti al valore assoluto di  F
```

Due esempi numerici aiuteranno il lettore all'uso della routine di calcolo:
sia da risolvere l'equazione algebrica di secondo grado

$$2x^2 - 3x + 7 = 0$$

impiegando il programma che abbiamo compilato:

```
            F5
a = ? 2
b = ? -3
c = ? 7
x1 = .75 + j 1.713914          x2 = .75 - j 1.713914

            F5
```

il calcolo mostra che le radici dell'equazione sono complesse coniugate; ciò dipende dal segno negativo del discriminante D:
$D = b^2 - 4ac = (-3)^2 - 4 \cdot 2 \cdot 7 = -47$.
Risolviamo ora l'equazione

$$-2x^2 - 3x + 7 = 0$$

```
            F5
a = ? - 2
b = ? - 3
c = ? 7
x1 = - 2.765564          x2 =  1.265564

            F5
```

i risultati mostrano che l'equazione ha due soluzioni reali; infatti il valore del discriminante è maggiore di zero:
$D = b^2 - 4ac = (-3)^2 - 4 \cdot (-2) \cdot 7 = 65$
La soluzione delle due equazioni proposte ha dato un'idea dell'utilità del programma, utilità che risulta ancor più evidente quando le equazioni da risolvere hanno coefficienti e termine noto formati da molte cifre, cosa che accade quasi sempre nelle applicazioni pratiche e che richiede operazioni lunghe e non prive di rischi d'errore; vediamone un caso per chiudere questo paragrafo; sia da risolvere l'equazione

$$-3.78442x^2 - .432776x + 2.88654 = 0$$

F5

a = ? -3.78442
b = ? -. 432776
c = ? 2.88654
x1 = - . 93240 x2 = . 8180427

F5

4.3.1 La soluzione grafica dell'equazione algebrica di secondo grado

La soluzione grafica dell'equazione di secondo grado non ha applicazioni pratiche ma, come le precedenti, è interessante perché aiuta a comprendere meglio ciò che sarà trattato nel prosieguo del testo. Prima di iniziare la procedura per il computo grafico delle radici dell'equazione è necessario ragionare sulla sua forma canonica:

$$a x^2 + b x + c = 0$$

il primo membro di questa espressione, come accennato nel paragrafo 2.28, rappresenta l'equazione di una parabola che il secondo membro vincola con y = 0; in altri termini si dice che l'equazione di secondo grado altro non è che la funzione parabola uguagliata a zero per la ricerca dei suoi punti di intersezione con l'asse delle ascisse.
Infatti data la parabola

$$y = a x^2 + b x + c$$

se questa interseca l'asse x in x1 ed x2, questi due valori coincidono con le radici dell'equazione sopra indicata.
A questo punto si devono ricordare alcune nozioni:

- Se la parabola interseca l'asse x, sia che rivolga la concavità verso l'alto, sia rivolga la concavità verso il basso, le due soluzioni reali dell'equazione di secondo grado, che si ottengono uguagliando a zero la funzione, sono le ascisse dei due punti .

- Se la parabola è soltanto tangente all'asse delle ascisse l'equazione che si ricava fornisce una sola radice (si dice in questo caso che le radici sono reali e coincidenti) che rappresenta l'ascissa dell'unico punto di contatto tra la parabola e l'asse x.

- Se la parabola non interseca l'asse x le soluzioni dell'equazione da essa ricavata non sono reali ma complesse coniugate e non forniscono alcuna informazione sulla geometria della curva.

Chiarito quanto sopra, determiniamo la soluzione grafica dell'equazione già sviluppata algebricamente nel paragrafo precedente:

$$2 x^2 - 3 x + 7 = 0$$

che trasformiamo nella funzione parabola corrispondente

$$y = 2 x^2 - 3 x + 7$$

a) in base alla natura della funzione si sceglie la presentazione cartesiana a 4 quadranti "sbloccando" la prima e la terza istruzione del programma di paragrafo 3.14
b) in base alla presentazione scelta si evidenziano il numero degli intervalli in cui sono divisi gli assi cartesiani: asse X = 10 intervalli asse Y = 10 intervalli
c) si fissano i valori del campo di variabilità della x in modo tale che la y corrispondente compia

certamente una escursione dai valori negativi a quelli positivi: ciò si ottiene, a vista, per x che varia da -10 a +10, con questi valori si ha un adattamento automatico al numero degli intervalli stabiliti per la calibrazione della scala
d) si riporta il valore massimo del campo di variabilità di y pari a quello della x = 10
e) si fissa l'incremento della variabile indipendente in .01
f) si riporta l'istruzione PSET relativa ai 4 quadranti **PSET (230 + k * x , 160 - k2 * y)**
g) si determina il coefficiente k in base al nuovo campo di variabilità fissato per la x in 10
k = 230 / 10 = 23
h) si determina il coefficiente k2 in base al nuovo semicampo fissato per y in 10
k2 = 160 / 10 = 16
i) sulla base dei punti f), g), h), si completa, in simbologia Qbasic, l'istruzione finale che ci consente di tracciare la nostra parabola su scala calibrata, si aggiunge il colore giallo:

PSET (230 + 23 * x , 160 - 16 * (2 * x ^ 2 - 3 * x +7)), 14

si compila infine il programma:

```
LINE ( 230, 0 )-( 230, 320 )  ' ASSE Y  4 quadranti

LINE ( 0, 160 )-( 460, 160 )  ' ASSE  X   2 e 4 quadranti

FOR  x = - 10  TO  10  STEP  .01  ' campo di variabilità ed incremento imposto per la x

PSET ( 230 +  23 * x , 160 -  16 * (  2 * x ^ 2 - 3 * x  + 7  ) ), 14  ' presentazione  della parabola

NEXT x   ' rimanda all'istruzione FOR x= -10 ecc.

LOCATE 23, 66: PRINT "y-Div.= 1"   ' intervalli scala asse Y  ;  1

LOCATE 22, 66: PRINT "x-Div.= 1"   ' intervalli scala asse X  ; 1

                F5
```

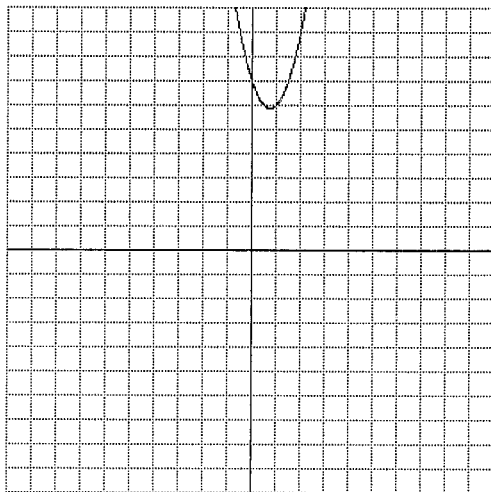



Figura 17
Grafico per la ricerca radici equazione di secondo grado (caso con radici complesse)
Campo di variabilità di x: da -10 a + 10
Scala asse x = 1 /div.
Scala asse y = 1/ div.

dal grafico risultante, riportato in figura 17, si osserva che la parabola non ha nessun punto di contatto con l'asse delle ascisse, ciò significa che l'equazione di secondo grado di cui cerchiamo le soluzioni non ha radici reali ma soltanto radici complesse; ciò peraltro era già stato calcolato algebricamente nel paragrafo 4.2. Nessun altro elemento dell'equazione data è ricavabile dal grafico.
Un secondo esempio ci mostra invece il caso in cui l'equazione di secondo grado ha radici reali; si prende in esame l'equazione

$$-2x^2 - 3x + 7 = 0$$ e si trasforma nella parabola corrispondente
$$Y = -2x^2 - 3x + 7$$

adottando l'identica procedura dell'esercizio precedente possiamo compilare direttamente il programma che sarà diverso dal primo soltanto per la nuova funzione parabola:

```
LINE ( 230, 0 )-( 230, 320 )  ' ASSE Y   4 quadranti

LINE ( 0, 160 )-(460, 160 )  ' ASSE X   2 e 4 quadranti

FOR  x = - 10  TO  10  STEP .01 ' campo di variabilità ed incremento imposto per la x

PSET ( 230 +  23 * x , 160 -  16  *  ( - 2 * x ^ 2 - 3 * x  + 7  ) ), 14  ' presentazione grafica della parabola

NEXT x  ' rimanda all'istruzione FOR x= -10 ecc.

LOCATE 23, 66: PRINT "y-Div.= 1"   ' intervalli scala asse Y  ; 1

LOCATE 22, 66: PRINT "x-Div.= 1"   ' intervalli scala asse X  ; 1
```

premendo F5 si ha il grafico risultante riportato in figura 18, si osserva che la parabola ha due punti di contatto con l'asse delle ascisse, ciò significa che l'equazione di secondo grado di cui cerchiamo le soluzioni ha due radici reali come era già stato calcolato algebricamente nel paragrafo 4.2. I valori approssimati rilevati a vista risultano $x1 = 1.2$ $x2 = -2.8$

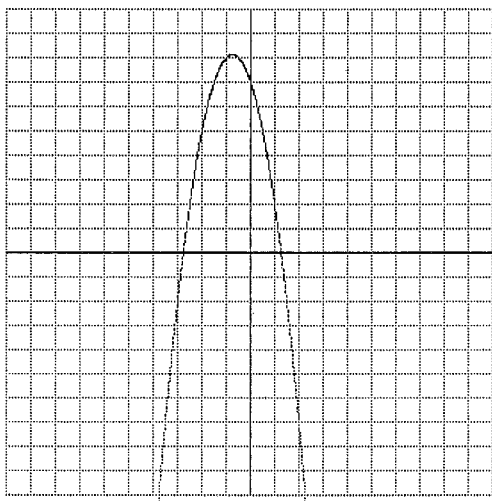



Figura 18
Grafico per ricerca radici
reali equazione secondo grado
Campo di variabilità di x:
da -10 a +10
Scala asse x = 1 /div.
Scala asse y = 1/ div.

4.4 Strumento grafico a scale variabili

E' illustrato in questo paragrafo uno strumento grafico indispensabile nella ricerca delle soluzioni delle equazioni algebriche di grado superiore al secondo e delle equazioni trascendenti. Lo strumento è derivato da quello impiegato fino ad ora e dà modo, a comando esterno, di variare rapidamente le scale del reticolo per la ricerca delle radici delle equazioni e la loro successiva determinazione numerica. Non conoscendo infatti la posizione grafica dei punti dell'asse x che risolvono l'equazione data è necessario, all'inizio della ricerca, allargare il campo di variabilità della x il più possibile per individuare di massima dove sono collocate le radici, una volta stabilita la zona interessata alle radici stesse si effettuano uno o più cambiamenti di scala in modo da poterne valutare i valori con la maggior precisione compatibile con la scala selezionata .

Il nuovo sistema grafico non è strutturato, come il precedente, per ottenere la riproducibilità delle funzioni in tutto il campo assegnato alla variabile indipendente, esso provoca il "troncamento" delle curve, al valore massimo del fondo scala delle ordinate, dove i tracciati non sono più utili per la ricerca delle radici dell'equazione da risolvere.

Questo strumento non fornisce alcuna indicazione in merito ad eventuali radici complesse.

Il programma deve eseguire i calcoli a doppia precisione per una corretta valutazione delle radici dell'equazione dopo la fase di ricerca.

Il programma che gestisce questa nuova grafica è qui mostrato compilato e commentato:

```
LINE ( 230, 0 )-( 230, 320 )  ' ASSE  Y 4 quadranti scelto per visualizzare qualsiasi tipo di funzione

LINE ( 0, 160 )-( 460, 160 )  ' ASSE  X 4 quadranti scelto per visualizzare qualsiasi tipo di funzione

LOCATE 10, 66: INPUT "x inizio" ; i   ' richiesta ingresso estremo inferiore campo variabilità di x

LOCATE 11, 66: INPUT "x fine"   ; f   ' richiesta ingresso estremo superiore campo variabilità di x

LOCATE 12, 66: INPUT "step"     ; p   ' richiesta ingresso valore dell'incremento di x (STEP)

LOCATE 13, 66: INPUT "x-Div."   ; s   ' richiesta ingresso valore intervallo reticolo asse x

LOCATE 14, 66: INPUT "y-Div"    ; t   ' richiesta ingresso valore intervallo reticolo asse y

LOCATE 15, 66: PRINT "-sn;+dx"   ' indicazione per l'istruzione successiva relativa allo spostamento
                                 ' a sinistra ( -sn) o a destra ( +dx) rispetto allo zero del tracciato
                                 ' della funzione

LOCATE 16, 66: INPUT "spost.x"  ; g   ' richiesta ingresso entità di spostamento del tracciato della
                                      ' funzione

FOR x = i  TO  f   STEP p   ' variazione della x nel campo stabilito a comando tra (inizio,i) e (fine,f)

y# =  f( x )  ' funzione ricavata dall'equazione da risolvere espressa a doppia precisione

r# = ( 16 / t )  * y#    ' calcolo variabile  r  a doppia precisione per ordinata istruzione PSET
                         ' comprensiva valore di scala stabilito a comando  per l'asse y ( lettera t)

IF r#  >  160 THEN r#  =  160  ' istruzione che evita l'overflow positivo durante la ricerca delle radici

IF r#  < - 160 THEN r#  =  -160 ' istruzione che evita l'overflow  negativo durante la ricerca delle radici

PSET ( 230 - ( 23 / s  ) * ( -g  ) + ( 23 / s  ) * x , 160 - r# ), 15  ' genera  il tracciato del grafico
             ' della funzione in base al valore di scala
             ' stabilito a comando  per l'asse x  (lettera s) , inoltre sposta il tracciato in base        &
```

' al valore assegnato a comando esterno da "spost.x" tramite la (lettera g)
NEXT x ' rimanda all'istruzione FOR x = i .....

Una precisazione deve essere fatta in merito alla scelta dei valori del campo di variabilità della x e delle scale da assegnare, sia alla variabile indipendente, sia alla variabile dipendente; non è sempre possibile al primo tentativo presentare la funzione nel migliore dei modi per poter ricavare le informazioni volute, a volte è necessario ripetere più di una volta l'impostazione grafica variando i valori introdotti nel programma. Negli esercizi che seguono, questo tipo di "messa a punto" del grafico è stato fatto, quando si è reso necessario, ma non lo si è menzionato per evitare al lettore pagine di descrizione inutili relative a tentativi preliminari che si risolvono in pochi minuti di lavoro al P.C .
La ricerca dei valori ottimali per una soddisfacente presentazione grafica delle funzioni si realizza dopo un poco di esperienza .

4.5 Soluzione dell'equazione algebrica di terzo grado

L'equazione algebrica di terzo grado, per la soluzione della quale sono state studiate diverse e complicate formule, può essere risolta facilmente mediante lo strumento grafico illustrato nel paragrafo 4.4. Vediamo per esempio come risolvere l'equazione:

$$-x^3 + x^2 + x + 1 = 0$$

per prima cosa se ne scrive la funzione corrispondente

$$y = -x^3 + x^2 + x + 1$$

in modo da poterne tracciare il grafico affinché questo mostri i punti in cui la curva interseca l'asse delle x:

-Se l'equazione ammette radici reali queste saranno individuate da detti punti, se l'equazione non ammette alcuna soluzione reale la curva non intercetterà l'asse delle ascisse.
-Per la ricerca delle radici è conveniente adottare un campo di variabilità della x piuttosto ampio, nel quale poter osservare come la variabile dipendente y si sviluppa allontanandosi decisamente dall'asse x.
-Dalla funzione corrispondente l'equazione data si vede facilmente che se fissiamo il campo di variabilità della x tra -10 (x inizio) e +10 (x fine) i valori che la y assume agli estremi del campo sono molto grandi e indicano una netta tendenza della funzione a " fuggire " da eventuali punti di contatto con l'asse x; ciò assicura che tale campo è adatto alla ricerca delle radici.
-Se nell'ambito del campo di variabilità della x vogliamo calcolare un numero sovrabbondante di punti, rispetto ai 460 pixel disponibili, possiamo assumere l'incremento della x pari a (step =.01), ottenendo in tal modo 20/.01 = 2000 punti di calcolo.
-Dato il campo di variabilità assunto per la x e considerato che il reticolo è formato da 20 intervalli è immediato fissare come valore di ciascuna divisione delle ascisse (Div.x= 1).
Ritenendo che quando la y raggiunge il valore 100 la funzione sia già in "fuga" stabiliamo con esso il massimo della scala delle ordinate assegnandole, per conseguenza, una suddivisione pari a (Div.y= 10).
-Essendo in fase di ricerca delle radici è necessario che la funzione venga presentata senza alcuna traslazione orizzontale; si porrà pertanto (spost. x = 0).

Determinati tutti i valori necessari alla nostra ricerca riscriviamo il programma di paragrafo 4.4 inserendovi la funzione da tracciare:

LINE (230, 0)-(230, 320) ' ASSE Y 4 quadranti scelti per visualizzare qualsiasi tipo di funzione &

```
LINE ( 0, 160 )-( 460, 160 )  ' ASSE  X 4 quadranti scelti per visualizzare qualsiasi tipo di funzione

LOCATE 10, 66: INPUT "x inizio"  ; i  ' richiesta ingresso estremo inferiore campo variabilità di x

LOCATE 11, 66: INPUT "x fine"    ; f  ' richiesta ingresso estremo superiore campo variabilità di x

LOCATE 12, 66: INPUT "step"      ; p  ' richiesta ingresso valore dell'incremento di x (STEP)

LOCATE 13, 66: INPUT "x-Div."    ; s  ' richiesta ingresso valore intervallo reticolo asse x

LOCATE 14, 66: INPUT "y-Div."    ; t  ' richiesta ingresso valore intervallo reticolo asse y

LOCATE 15, 66: PRINT "-sn;+dx"  ' indicazione per l'istruzione successiva relativa allo spostamento
                                ' a sinistra ( -sn) o a destra ( +dx) rispetto allo zero del tracciato
                                ' della funzione

LOCATE 16, 66: INPUT "spost.x"  ;  g  ' richiesta ingresso entità di spostamento del tracciato della
                                      ' funzione

FOR x = i  TO  f   STEP  p  '  variazione della x nel campo stabilito a comando tra (inizio,i) e (fine,f)

y# = - x ^ 3 + x ^ 2 + x + 1  ' funzione ricavata dall'equazione da risolvere espressa a doppia
                              ' precisione

r# = ( 16 / t ) * y#  ' calcolo variabile r a doppia precisione per ordinata istruzione PSET
                      ' comprensiva valore di scala stabilito a comando per l'asse y ( lettera t)

IF r#  > 160 THEN  r# = 160 ' istruzione che evita l'overflow positivo durante la ricerca delle radici

IF r# < - 160 THEN  r# = - 160 ' istruzione che evita l'overflow negativo durante la ricerca delle radici

PSET ( 230 - ( 23 / s ) * ( -g ) + ( 23 / s ) * x , 160 - r# ), 15  ' genera il tracciato del grafico
                                                                    ' della funzione in base al valore di scala
            ' stabilito a comando per l'asse x (lettera s) , inoltre sposta il tracciato in base
            ' al valore assegnato a comando esterno da "spost.x" tramite la (lettera g)
            ' rimanda all'istruzione FOR x = i .....

NEXT x  ' rimanda all'istruzione FOR x = i ....
              F5
```

dopo aver digitato F5 si ottiene, a sinistra dello schermo, il reticolo libero e sulla destra la prima richiesta dati che completata da seguito alla seconda e così via, dove digitiamo i valori che abbiamo fissato in precedenza come sotto riportato:

```
x inizio? -10
x fine? +10
step.? .01
x-Div:? 1
y-Div.? 10
-sn ; +dx
spost. x ? 0
```

dopo l'introduzione dell'ultimo dato si ha la presentazione del grafico riportato in figura 19.

Dal tracciato si osserva:
-per valori di x < - 4.2 la y assume il valore costante di fondo scala positivo pari a +100, ciò dà luogo ad un segmento orizzontale posto tutto in alto a sinistra del reticolo (condizione imposta dal programma per evitare l'overflow). In questa zona in realtà la funzione è in "fuga".

-per valori di x > 5 la y assume il valore costante di fondo scala negativo pari a -100, ciò dà luogo ad un segmento orizzontale posto tutto in basso a destra del reticolo (condizione imposta dal programma per evitare l'overflow). In questa zona in realtà la funzione è in "fuga".
-per valori di x compresi tra - 4.2 e +5 la curva taglia l'asse delle ascisse in un solo punto, ad un valore di x valutabile con approssimazione a circa 1.8; questo rilievo ci dice che l'equazione ha una sola radice reale computabile al momento in x1 = 1.8.
Individuata l'unica radice reale dell'equazione si può procedere ad una sua valutazione più precisa mediante cambiamenti dei valori introdotti per la ricerca .

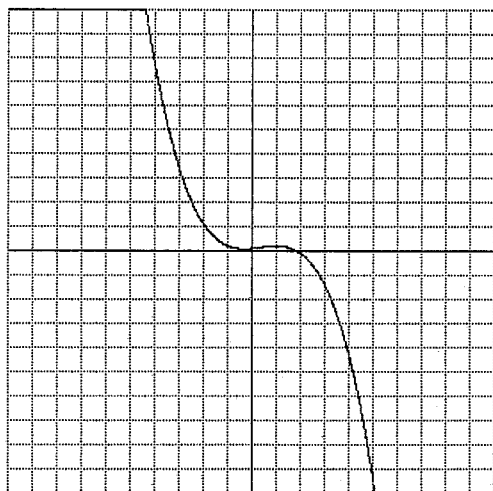



Figura 19
Grafico ricerca radici equazione algebrica di terzo grado
Campo di variabilità della x:
da +10 a -10
Scala asse x = 1/div.
Scala asse y = 10/div.

Procediamo ad una migliore valutazione della radice in base a queste considerazioni:
-Dato che la radice è stata individuata nell'intorno di 1.8 fissiamo un nuovo campo di variabilità della x attorno a tale valore; tra 1.7 e 1.9 . (x inizio = 1.7) (x fine = 1.9)
-Fissiamo un incremento pari a .0001 al quale corrispondono (1.9 - 1.7) / .0001 = 2000 punti di calcolo (step = .0001)
-Determiniamo l'intervallo di divisione orizzontale del reticolo in (1.9 -1.7) / 10 = .02 (Div-x=.02) dal quale si ottiene una precisione di lettura migliore di 2/100.
-Per consentire un più facile apprezzamento dell'intersezione della curva con l'asse delle ascisse si fissa la scala delle ordinate a .1 per divisione (Div-Y=.1)
-Si impone alla curva uno spostamento verso sinistra pari a 1.7 (spost.x = -1.7) in modo che l'origine degli assi cartesiani non ha più il valore di riferimento 0 per le ascisse ma assume il valore +1.7
Il valore della radice che si leggerà sarà pertanto la somma tra il valore fisso 1.7 e il numero di divisioni che separano il punto di misura dall'origine degli assi.
Si ripete l'elaborazione a programma con i nuovi dati: F5

```
x inizio? + 1.7
x fine? +1. 9
step.? .0001
x-Div:? .02
y-Div.? .1
-sn ; +dx
spost. x ? -1.7
```

si ottiene il tracciato di figura 20 in cui è presentata soltanto una piccola parte della funzione nell'intorno del suo punto di intersezione con l'asse x.

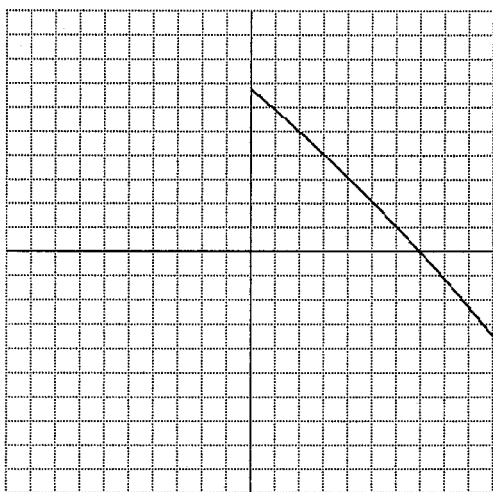



Figura 20
Grafico per valutazione radici equazione algebrica di terzo grado
Campo di variabilità della x:
da +1.7 a +1.9
Scala asse x = .02 / div.
Scala asse y = .1 / div.

Dal tracciato si osserva:

-soltanto una piccola parte della funzione è tracciata tra il primo e il quarto quadrante
-il campo in cui la funzione è presentata è definito esclusivamente tra l'origine degli assi, che ora rappresenta il valore + 1.7, e l'estremo del campo di variabilità di x fissato in +1.9.
-il tratto di curva taglia l'asse delle ascisse a circa 7 divisioni rispetto all'origine degli assi; cioè ad una distanza dall'origine pari a $7 \cdot .02 = .14$.

Si conclude che il valore della radice è $x1 = + 1.7 + .14 = 1.84$
Se si desidera una precisione maggiore nella determinazione del valore della radice si ripete il procedimento impiegando un campo di variabilità della x più stretto del precedente con incrementi delle scale del reticolo e spostamento della curva, opportunamente adattati alle esigenze di misura.
Questi risultati, che si possono ottenere in pochi minuti di lavoro, mostrano l'indubbia utilità ed efficacia del metodo grafico che abbiamo impiegato.

Si consiglia il lettore di tentare la soluzione di altre equazioni algebriche, sia di terzo grado, sia di grado superiore al terzo, per prendere pratica con la procedura di calcolo.

4.6 Le equazioni trascendenti

Si dicono equazioni trascendenti tutte quelle che non sono algebriche; ad esempio l'equazione

$$\mathrm{Log}\, x = 0$$

è una equazione trascendente, la soluzione della quale si ricava immediatamente mediante la corrispondente potenza in base dieci

$$x = 10^0 = 1$$

Le equazioni trascendenti si possono dividere in "due tipi":

-equazioni che con opportune manipolazioni matematiche permettono di ottenere le soluzioni; tali tipi sono prevalentemente riportati nei libri di testo come utili esercitazioni sull'argomento ma purtroppo in pratica si riscontrano assai di rado

-equazioni trascendenti scaturite da un problema reale non risolvibili, purtroppo, mediante eleganti giochi formali.

Se ci proponiamo ad esempio la soluzione della semplice equazione trascendente

$$e^x - 10 (x)^{1/2} = 0$$

non abbiamo altra strada che tentarne la soluzione grafica nel seguente modo:

-si compone la funzione trascendente, di cui si traccerà il grafico, partendo dall'equazione data

$$y = e^x - 10 (x)^{1/2}$$

-si scrive la procedura di impostazione
-Per la ricerca delle radici è necessario adottare un campo di variabilità della x contenuto nel semicampo dei valori positivi dato che la funzione ha la variabile indipendente sotto radice quadrata.
-Dalla funzione corrispondente l'equazione data si vede facilmente che se fissiamo il campo di variabilità della x tra 0 (x inizio) e +10 (x fine) il valore che la y assume per x = 10 è molto grande e indica una netta tendenza della funzione a " fuggire " da eventuali punti di contatto con l'asse x; ciò assicura che tale campo è adatto alla ricerca delle radici.
-Se nell'ambito del campo di variabilità della x vogliamo calcolare un numero sovrabbondante di punti, rispetto ai 230 pixel disponibili, possiamo assumere l'incremento della x pari a (step =.01), ottenendo in tal modo 10 / .01 = 1000 punti di calcolo.
-Dato il semicampo di variabilità assunto per la x e considerato che il reticolo è formato da 10 intervalli è immediato fissare come valore di ciascuna divisione delle ascisse (Div.x= 1).
Ritenendo che quando la y raggiunge il valore 100 la funzione sia già in "fuga" stabiliamo con esso il massimo della scala delle ordinate assegnandole, per conseguenza, una suddivisione pari a (Div.y=10).
-Essendo in fase di ricerca delle radici è necessario che la funzione venga presentata senza alcuna traslazione orizzontale; si porrà pertanto (spost. x = 0).

Determinati tutti i valori necessari alla nostra ricerca riscriviamo il programma di paragrafo 4.4 inserendovi la funzione da tracciare:

```
LINE ( 230, 0 )-( 230, 320 )  ' ASSE  Y  4 quadranti scelti per visualizzare qualsiasi tipo di funzione

LINE ( 0, 160 )-( 460, 160 )  ' ASSE  X  4 quadranti scelti per visualizzare qualsiasi tipo di funzione

LOCATE 10, 66: INPUT "x inizio"  ;  i  ' richiesta ingresso estremo inferiore campo variabilità di x

LOCATE 11, 66: INPUT "x fine"    ;  f  ' richiesta ingresso estremo superiore campo variabilità di x

LOCATE 12, 66: INPUT "step"      ;  p  ' richiesta ingresso valore dell'incremento di x (STEP)          &
```

```
LOCATE 13, 66: INPUT "x-Div."  ; s  ' richiesta ingresso valore intervallo reticolo asse x

LOCATE 14, 66: INPUT "y-Div."  ; t  ' richiesta ingresso valore intervallo reticolo asse y

LOCATE 15, 66: PRINT "-sn;+dx"  ' indicazione per l'istruzione successiva relativa allo spostamento
                                ' a sinistra ( -sn) o a destra ( +dx) rispetto allo zero del tracciato
                                ' della funzione

LOCATE 16, 66: INPUT "spost.x"  ;  g  ' richiesta ingresso entità di spostamento del tracciato della
                                      ' funzione

FOR  x = i  TO  f  STEP  p  ' variazione della x nel campo stabilito a comando tra (inizio,i) e (fine,f)

y# = EXP ( x ) - 10 * SQR ( x )  ' funzione ricavata dall'equazione da risolvere espressa a doppia
                                 ' precisione

r# = ( 16 / t ) * y#  ' calcolo variabile r a doppia precisione per ordinata istruzione PSET
                      ' comprensiva valore di scala stabilito a comando per l'asse y ( lettera t)

IF r# > 160 THEN r# = 160  ' istruzione che evita l'overflow positivo durante la ricerca delle radici

IF r# < -160 THEN r# = -160 ' istruzione che evita l'overflow negativo durante la ricerca delle radici

PSET ( 230 - ( 23 / s ) * ( -g ) + ( 23 / s ) * x , 160 - r# ), 15  ' genera il tracciato del grafico
                    ' della funzione in base al valore di scala
                    ' stabilito a comando per l'asse x (lettera s) , inoltre sposta il tracciato in base
                    ' al valore assegnato a comando esterno da "spost.x" tramite la (lettera g)

NEXT x  ' rimanda all'istruzione FOR x = i .....
```

F5

dopo aver digitato F5 si ottiene, a sinistra dello schermo, il reticolo libero e sulla destra la prima richiesta dati che completata dà seguito alla seconda e così via, dove digitiamo i valori che abbiamo fissato in precedenza come sotto riportato:

```
x inizio? 0
x fine? +10
step.? .01
x-Div:? 1
y-Div.? 10
-sn ; +dx
spost. x ? 0
```

dopo l'introduzione dell'ultimo dato si ha la presentazione del grafico riportato in figura 21 .
Dal tracciato si osserva:
-per valori di x > 4.7 la y assume il valore costante di fondo scala positivo pari a +100, ciò dà luogo ad un segmento orizzontale posto tutto in alto a destra del reticolo (condizione imposta dal programma per evitare l'overflow). In questa zona in realtà la funzione è in "fuga".
-per valori di x compresi tra 0 e 4.7 la curva taglia l'asse delle ascisse, apparentemente, in un solo punto, ad un valore di x valutabile con approssimazione a circa 2.8; questo rilievo ci dice che l'equazione ha una radice reale $x1 = 2.8$.
-la curva mostra inoltre valori negativi della y nell'intervallo della x compreso tra 0 e 2.8, cosa apparentemente in contrasto con il fatto che la funzione vale $y = 1$ per $x = 0$, ciò denuncia un passaggio del tracciato dai valori positivi ai valori negativi nelle immediate vicinanze dell'origine degli assi; a questo passaggio corrisponde un punto la cui ascissa, x2, è la seconda radice

dell'equazione data. Questa radice, non esplicitamente evidenziata, deve essere cercata con gli opportuni cambiamenti di scala.

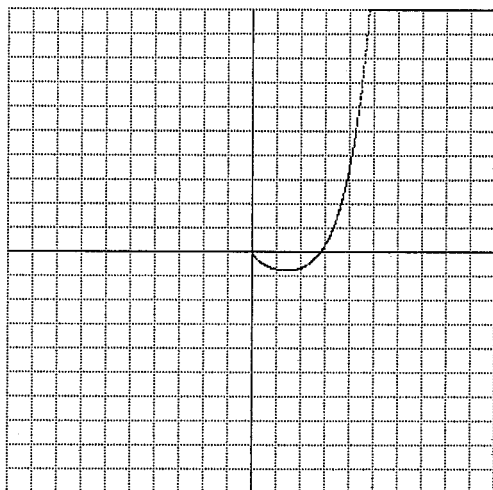



Figura 21
Grafico per ricerca radici equazione trascendente
Campo di variabilità della x: da 0 a + 10
Scala asse x = 1 / div.
Scala asse y = 10 / div.

Procediamo ad una migliore valutazione della radice x1 in base alle seguenti considerazioni:

-Dato che la radice è stata individuata nell'intorno di 2.8 fissiamo un nuovo campo di variabilità della x attorno a tale valore; tra 2 e 3 . (x inizio = 2) (x fine = 3)
-Fissiamo un incremento pari a .001 al quale corrispondono (3 - 2) / .001 = 1000 punti di calcolo (step = .001)
-Determiniamo l'intervallo di divisione orizzontale del reticolo in (3 - 2) / 10 = .1 (Div-x=.1) dal quale si ottiene una precisione di lettura migliore di 1/10.
-Per un più facile apprezzamento dell'intersezione della curva con l'asse delle ascisse si fissa la scala delle ordinate a .1 per divisione (Div-Y=.1)
-Si impone alla curva uno spostamento verso sinistra pari a 2 (spost.x = -2) in modo che l'origine degli assi cartesiani non ha più il valore di riferimento 0 per le ascisse ma assume il valore + 2. Il valore della radice che si leggerà sarà pertanto la somma tra il valore fisso 2 e il numero di divisioni che separano il punto di misura dall'origine degli assi.

Si ripete l'elaborazione a programma con i nuovi dati:

```
F5
                         x inizio? + 2
                         x fine? + 3
                         step.? .001
                         x-Div:? .1
                         y-Div.? .1
                         -sn ; +dx
                         spost. x ? - 2
```

si ottiene il tracciato di figura 22 in cui è presentata soltanto una parte della funzione nell'intorno del suo punto di intersezione con l'asse x.

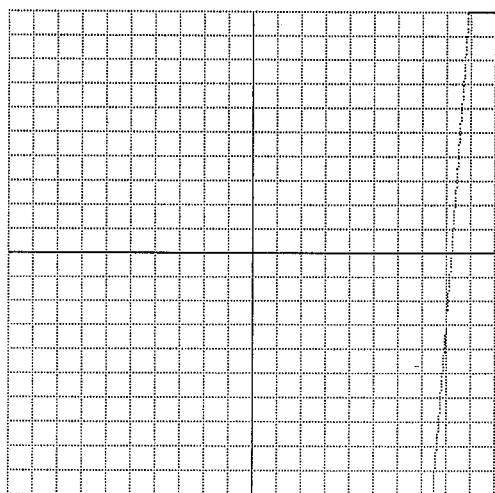



Figura 22
Grafico per valutazione prima radice equazione trascendente
Campo di variabilità della x: da 2 a 3
Scala asse x = .1 / div.
Scala asse y = .1 / div.

Dal tracciato si osserva:

-soltanto una parte della funzione è tracciata tra il primo e il quarto quadrante
-il campo in cui la funzione è presentata è definito esclusivamente tra l'origine degli assi, che ora rappresenta il valore + 2, e l'estremo del campo di variabilità di x fissato in +3
-il tratto di curva taglia l'asse delle ascisse a circa 8.2 divisioni rispetto all'origine degli assi; cioè ad una distanza dall'origine pari a $8.2 \cdot .1 = .82$.
Si conclude che il valore della radice è $x1 = + 2 + .82 = 2.82$

Se si desidera una precisione maggiore nella determinazione del valore della radice x1 si ripete il procedimento impiegando un campo di variabilità della x più stretto del precedente con incrementi delle scale del reticolo e spostamento della curva, opportunamente adattati alle esigenze di misura.
Si esegue ora la ricerca e la valutazione della radice x2 che, come abbiamo visto, deve trovarsi vicino all'origine degli assi:

-Dato che la radice è stata individuata indirettamente vicino allo 0 fissiamo un nuovo campo di variabilità della x tra 0 e .1 (x inizio = 0) (x fine = .1)
-Fissiamo un incremento pari a .0001 al quale corrispondono (.1 - 0) / .0001 = 1000 punti di calcolo (step = .0001)
-Determiniamo l'intervallo di divisione orizzontale del reticolo in (.1 - 0) / 10 = .01 (Div-x=.01) dal quale si ottiene una precisione di lettura migliore di 1/100.
-Per ottenere un più facile apprezzamento dell'intersezione della curva con l'asse delle ascisse si fissa la scala delle ordinate a .1 per divisione (Div-Y=.1)
-Dato che non è necessario spostare la curva sarà (spost.x = 0)

Si ripete l'elaborazione a programma con i nuovi dati:

F5

x inizio? 0
x fine? +.1
step.? .0001
x-Div:? .01
y-Div.? .1
-sn ; +dx
spost. x ? 0

si ottiene il tracciato di figura 23 in cui è presentata soltanto una parte della funzione nell'intorno del suo punto di intersezione con l'asse x.

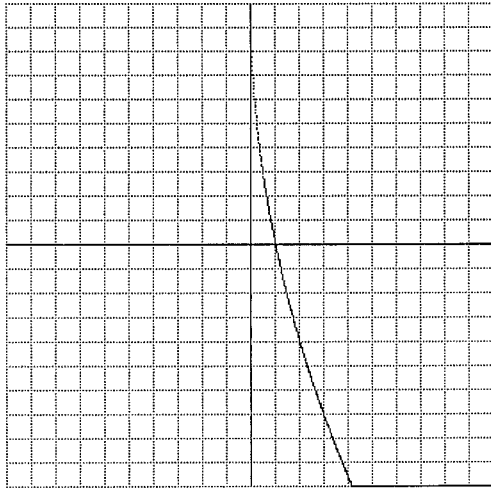



Figura 23
Grafico per valutazione seconda radice equazione trascendente
Campo di variabilità della x : da 0 a .1
Scala asse x = .01 / div.
Scala asse y = .1 / div.

Dalla figura 23 si osserva:

-soltanto una parte della funzione è tracciata tra il primo e il quarto quadrante
-in questa curva si evidenzia chiaramente l'intersezione della stessa con l'asse delle ascisse, cosa che avevamo soltanto dedotto, in precedenza, sulla base di certe osservazioni.
-il campo in cui la funzione è presentata è definito esclusivamente tra l'origine degli assi e l'estremo del campo di variabilità di x fissato in +.1
-il tratto di curva taglia l'asse delle ascisse ad una divisione rispetto all'origine degli assi, cioè ad un valore pari a .01; si conclude che il valore della radice è x2 = .01.

Se si desidera una precisione maggiore nella determinazione del valore della radice x2 si ripete il procedimento impiegando un campo di variabilità della x più stretto del precedente con incrementi, scale del reticolo, spostamento della curva, opportunamente adattati alle esigenze di misura.
Riassumendo i risultati del lavoro svolto possiamo concludere che l'equazione trascendente

$$e^{x} - 10 (x)^{1/2} = 0$$

ha due radici reali così definite:
x1 = 2.82; x2 = .01 .

4.7 I sistemi di equazioni trascendenti

Il sistema grafico impiegato per la soluzione delle equazioni trascendenti è utilizzabile anche per trovare le soluzioni di sistemi di tali equazioni. Vediamo in quale modo si deve procedere per questo tipo di applicazione mediante il seguente esempio:
sia da risolvere il sistema trascendente

$$\begin{cases} y - \text{Sen}\, x - 2 = 0 \\ e^x - .2\,y = 0 \end{cases}$$

per prima cosa si esplicitano le y

$$\begin{cases} y = \text{Sen}\, x + 2 \\ y = 5\, e^x \end{cases}$$

dopo di che si procede ad eguagliare i secondi membri del sistema come se si dovesse risolverlo in modo convenzionale

$$\text{Sen}\, x + 2 = 5\, e^x$$

per scrivere

$$\text{Sen}\, x + 2 - 5\, e^x = 0$$

a questo punto, dal sistema dato, siamo passati ad una equazione che possiamo risolvere con lo stesso metodo applicato nel paragrafo 4.6 mediante la stesura della funzione corrispondente:

$$y = \text{Sen}\, x + 2 - 5\, e^x$$

-Per la ricerca delle radici è necessario adottare un campo di variabilità piuttosto ampio onde trovare con più probabilità le intersezioni della funzione con l'asse delle x.
-Dalla funzione corrispondente l'equazione data si vede facilmente che se fissiamo il campo di variabilità della x tra -10 (x inizio) e +10 (x fine) i valori che la y assume all'estremo inferiore sono governati da Sen x + 2 che è sempre positivo e non interseca l'asse x, mentre per l'estremo superiore sono governati da -5 exp x che porta la funzione in fuga verso i valori negativi; è quindi certo che nel campo di variabilità scelto si troveranno le radici cercate.
-Se nell'ambito del campo di variabilità della x vogliamo calcolare un numero sovrabbondante di punti, rispetto ai 460 pixel disponibili, possiamo assumere l'incremento della x pari a (step =.01), ottenendo in tal modo 20 / .01 = 2000 punti di calcolo.
-Dato il campo di variabilità assunto per la x e considerato che il reticolo è formato da 20 intervalli è immediato fissare come valore di ciascuna divisione delle ascisse (Div.x= 1).
-Ritenendo che quando la y raggiunge il valore -10 la funzione sia già in "fuga" stabiliamo la scala delle ordinate assegnandole una suddivisione pari a (Div.y=1).
-Essendo in fase di ricerca delle radici è necessario che la funzione venga presentata senza alcuna traslazione orizzontale; si porrà pertanto (spost. x = 0).
Determinati tutti i valori necessari alla nostra ricerca riscriviamo il programma di paragrafo 4.4 inserendovi la funzione da tracciare:

```
LINE ( 230, 0 )-( 230, 320 )  ' ASSE  Y 4  quadranti scelti per visualizzare qualsiasi tipo di funzione

LINE ( 0, 160 )-( 460, 160 )  ' ASSE  X 4  quadranti scelti per visualizzare qualsiasi tipo di funzione        &
```

```
LOCATE 10, 66: INPUT "x inizio"  ; i  ' richiesta ingresso estremo inferiore campo variabilità di x

LOCATE 11, 66: INPUT "x fine"    ; f  ' richiesta ingresso estremo superiore campo variabilità di x

LOCATE 12, 66: INPUT "step"      ; p  ' richiesta ingresso valore dell'incremento di x (STEP)

LOCATE 13, 66: INPUT "x-Div."    ; s  ' richiesta ingresso valore intervallo reticolo asse x

LOCATE 14, 66: INPUT "y-Div."    ; t  ' richiesta ingresso valore intervallo reticolo asse y

LOCATE 15, 66: PRINT "-sn;+dx"      ' indicazione per l'istruzione successiva relativa allo spostamento
                                    ' a sinistra ( -sn) o a destra ( +dx) rispetto allo zero del tracciato
                                    ' della funzione
LOCATE 16, 66: INPUT "spost.x"  ; g  ' richiesta ingresso entità di spostamento del tracciato della
                                     ' funzione

FOR  x = i  TO  f  STEP  p  '  variazione della x nel campo stabilito a comando tra (inizio,i) e (fine,f)

y# = SIN ( x ) + 2 - 5 * EXP ( x ) ' funzione ricavata dal sistema da risolvere espressa a doppia precisione

r# = ( 16 / t ) * y#  ' calcolo variabile r a doppia precisione per ordinata istruzione PSET
                      ' comprensiva valore di scala stabilito a comando per l'asse y ( lettera t)

IF r# > 160 THEN r# = 160 ' istruzione che evita l'overflow positivo durante la ricerca delle radici

IF r# < -160 THEN r# = - 160 ' istruzione che evita l'overflow negativo durante la ricerca delle radici

PSET ( 230 - ( 23 / s ) * ( -g ) + ( 23 / s ) * x , 160 - r# ), 15 ' genera il tracciato del grafico
                                ' della funzione in base al valore di scala
                                ' stabilito a comando per l'asse x (lettera s) , inoltre sposta il tracciato in base
                                ' al valore assegnato a comando esterno da "spost.x" tramite la (lettera g)

NEXT x  ' rimanda all'istruzione FOR x = i .....

            F5
```

dopo aver digitato F5 si ottiene, a sinistra dello schermo, il reticolo libero e sulla destra la prima richiesta dati che completata dà seguito alla seconda e così via, dove digitiamo i valori che abbiamo fissato in precedenza come sotto riportato:

```
x inizio? -10
x fine? +10
step.? .01
x-Div:? 1
y-Div.? 1
-sn ; +dx
spost. x ? 0
```

dopo l'introduzione dell'ultimo dato si ha la presentazione del grafico riportato in figura 24 .

Dal tracciato si osserva:
-per valori di x < -1.6 la y ondula secondo Sen x nel campo delle y positive
-per valori di x compresi tra -1.8 e +1 la curva taglia l'asse delle ascisse in un solo punto, ad un valore di x valutabile con approssimazione a circa -1.6; questo rilievo ci dice che l'equazione ha una radice reale x1 = -1.6
-la curva mostra inoltre la fuga della funzione, nel campo negativo di y, nell'intervallo della x compreso tra 1 e 10.

Procediamo ad una migliore valutazione della radice x1 in base alle seguenti considerazioni:

-Dato che la radice è stata individuata nell'intorno di -1.6 fissiamo un nuovo campo di variabilità della x attorno a tale valore; tra -1.7 e -1.5 . (x inizio = -1.7) (x fine = -1.5)
-Fissiamo un incremento pari a .0001 al quale corrispondono (1.7-1.5) / .0001 = 2000 punti di calcolo (step = .0001)
-Determiniamo l'intervallo di divisione orizzontale del reticolo in (1.7-1.5) / 10 = .02 (Div-x=.02) dal quale si ottiene una precisione di lettura migliore di 2/100.
-Per ottenere un più facile apprezzamento dell'intersezione della curva con l'asse delle ascisse si fissa la scala delle ordinate a .01 per divisione (Div-Y=.01)
-Si impone alla curva uno spostamento verso destra pari a 1.7 (spost.x = +1.7) in modo che l'origine degli assi cartesiani non ha più il valore di riferimento 0 per le ascisse ma assume il valore -1.7
I valore della radice che si leggerà sarà pertanto la somma algebrica tra il valore fisso -1.7 e il numero di divisioni che separano il punto di misura dall'origine degli assi.

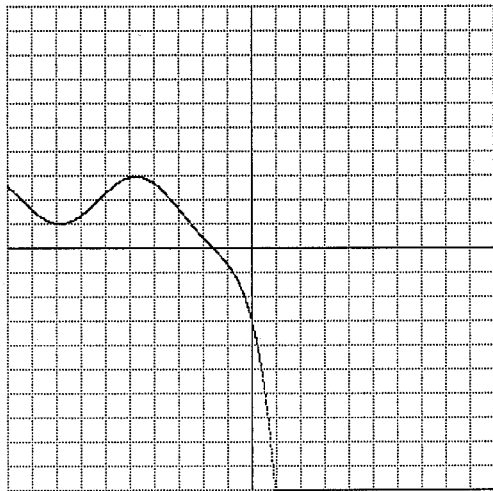



Figura 24
Grafico per ricerca soluzione sistema trascendente
Campo di variabilità della x :
da -10 a + 10
Scala asse x = 1 / div.
Scala asse y = 1 / div.

Si ripete l'elaborazione a programma con i nuovi dati:

F5

```
x inizio? -1.7
x fine?  -1.5
step.? .0001
x-Div:? .02
y-Div.? .01
-sn ; +dx
spost. x ? +1.7
```

si ottiene il tracciato di figura 25 in cui è presentata soltanto una parte della funzione nell'intorno del suo punto di intersezione con l'asse x.

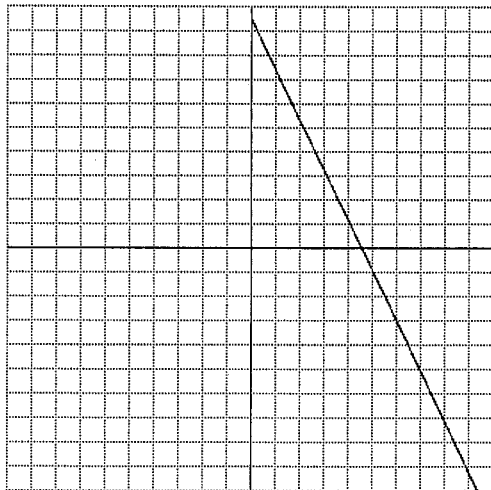



Figura 25
Grafico per soluzione ad alta precisione sistema trascendente
Campo di variabilità della x: da -1.7 a - 1.5
Scala asse x = .02/div.
Scala asse y =.01/div.

Dal tracciato si osserva:

-soltanto una parte della funzione è tracciata tra il primo e il quarto quadrante
-il campo in cui la funzione è presentata è definito esclusivamente tra l'origine degli assi, che ora rappresenta il valore - 1.7, e l'estremo del campo di variabilità di x fissato in -1.5
-il tratto di curva taglia l'asse delle ascisse a circa 4.5 divisioni rispetto all'origine degli assi; cioè ad una distanza dall'origine pari a $4.5 \cdot .02 = .09$
si conclude che il valore della radice è $x1 = -1.7 + .09 = -1.61$
Per completare la soluzione del sistema dato dobbiamo infine calcolare il valore di y1 corrispondente ad x1 sulla base di una delle due equazioni che lo compongono:

$$y = \text{Sen } x + 2 = \text{Sen } x1 + 2 = \text{Sen } (-1.61) + 2 = 1.0007$$

la soluzione completa del sistema è pertanto:

$$x1 = -1.61$$
$$y1 = 1.0007$$

4.8 Osservazioni sull'impiego del mezzo grafico

Concludiamo questo capitolo con alcune osservazioni sull'impiego del mezzo grafico.
Gli schemi di lavoro che ci hanno dato la possibilità di risolvere alcune equazioni sono stati adattati dall'autore, dopo alcune prove, alle diverse funzioni da trattare; gli schemi non sono rigidi, con un poco di esperienza si possono compilare schemi diversi ed, in brevissimo tempo, eseguire con essi numerosi tentativi con il P.C. per individuare al meglio i campi di variabilità della x che consentono di evidenziare le intersezioni del grafico della funzione con l'asse delle ascisse. Le potenzialità del sistema grafico potranno essere ulteriormente sviluppate dal lettore aggiungendo e, o, modificando il programma originale nel modo ritenuto più adatto alle proprie esigenze.

CAPITOLO 5
LA RICERCA DEI PUNTI NOTEVOLI DELLE FUNZIONI

Il capitolo 5 è dedicato all'impiego del Qbasic per la ricerca dei punti notevoli delle funzioni; luoghi di massimo e minimo relativi che caratterizzano le funzioni stesse.

5.1 Il problema della ricerca dei massimi e minimi di una funzione

E' noto come si possano determinare i punti di massimo e minimo relativi di una funzione quando questa sia facilmente derivabile; si ripropone il procedimento analitico a scopo di rinfrescare quanto appreso durante gli studi scolastici:

data la funzione $y = x^2 + x + 1$

se ne vogliono determinare eventuali punti di massimo o di minimo; a tale scopo se ne calcolano la derivata prima e la derivata seconda

$$y' = 2x + 1$$

$$y'' = 2$$

si eguaglia a zero la derivata prima risolvendo l'equazione che si ottiene

$$y' = 2x + 1 = 0$$

la soluzione di $2x + 1 = 0$ si ha per $x1 = -.5$; questa radice denuncia la presenza di un punto notevole la cui natura è ancora da stabilire.
Per conoscere se il punto è un massimo od un minimo della funzione si deve valutare che segno assume la derivata seconda in corrispondenza di $x1 = -.5$; questa sostituzione non è possibile dato che la y" è, in questo caso, una costante indipendente da x. Si considera allora il segno di y" così come è scaturito dal calcolo ($y'' = 2$); essendo $y'' > 0$, si può affermare che la funzione data ha un minimo per $x1 = -.5$.
L'esempio che abbiamo fatto ha mostrato l'estrema semplicità di tale operazione che richiede peraltro modestissime conoscenze di analisi matematica.
Le difficoltà che si incontrano nella pratica corrente però non permettono, il più delle volte, la soluzione del problema in termini così piani, sia a causa delle difficoltà di calcolo delle derivate se le funzioni sono molto complicate, sia per la difficoltà di soluzione delle equazioni discendenti dalle derivate, qualora le stesse siano state calcolate.
E' possibile risolvere molti problemi per la ricerca dei punti notevoli di una funzione, senza fare ricorso a mezzi di analisi matematica, ricorrendo, tramite il Qbasic, all'aiuto dei sistemi grafici con il supporto della programmazione numerica introdotta nel capitolo 2.
Le esercitazioni che seguono mostrano questa metodologia "mista" che dà interessanti risultati sul piano pratico.

5.2 Esercitazione grafico-numerica n° 1

Sia data la funzione

$$y = x^4 + x^3 + x^2 + x + 1$$

se ne vogliano ricercare i punti notevoli.
Se dovessimo risolvere il problema per via puramente analitica il calcolo delle derivate sarebbe semplice, difficile sarebbe invece la ricerca algebrica delle radici della equazione discendente dalla derivata prima che, come si intuisce, è una equazione completa di terzo grado.
E' chiaro che la strada da percorrere è quella indicata nel paragrafo 5.1 secondo la quale si devono impiegare metodi grafici abbinati a computazioni numeriche; seguiamo pertanto uno schema preliminare simile a quello adottato per l'impiego dello strumento grafico di paragrafo 4.4:

-Per la ricerca dei punti notevoli è necessario adottare un campo di variabilità della x piuttosto ampio onde poterli individuare.
-Si fissa il campo di variabilità della x tra -10 (x inizio) e +10 (x fine) dato che per questi due valori estremi la funzione è già in "fuga" .
-Se nell'ambito del campo di variabilità della x vogliamo calcolare un numero sovrabbondante di punti, rispetto ai 460 pixel disponibili, possiamo assumere l'incremento della x pari a (step =.01), ottenendo in tal modo 20 / .01 = 2000 punti di calcolo.
-Dato il campo di variabilità assunto per la x e considerato che il reticolo è formato da 20 intervalli, è immediato fissare come valore di ciascuna divisione delle ascisse (Div.x= 1).
-Ritenendo che quando la y raggiunge il valore +10 o -10 la funzione sia già in "fuga", stabiliamo la scala delle ordinate assegnandole una suddivisione pari a (Div.y=1).
-Essendo in fase di ricerca dei punti notevoli è necessario che la funzione venga presentata senza alcuna traslazione orizzontale; si porrà pertanto (spost. x = 0).

Determinati tutti i valori necessari alla nostra ricerca riscriviamo il programma di paragrafo 4.4 inserendovi la funzione da studiare:

```
LINE ( 230, 0 )-( 230, 320 )  ' ASSE  Y 4 quadranti scelti per visualizzare qualsiasi tipo di funzione

LINE ( 0, 160 )-( 460, 160 )  ' ASSE  X 4 quadranti scelti per visualizzare qualsiasi tipo di funzione

LOCATE 10, 66: INPUT "x inizio"  ;  i  ' richiesta ingresso estremo inferiore campo variabilità di x

LOCATE 11, 66: INPUT "x fine"    ;  f  ' richiesta ingresso estremo superiore campo variabilità di x

LOCATE 12, 66: INPUT "step"      ;  p  ' richiesta ingresso valore dell'incremento di x (STEP)

LOCATE 13, 66: INPUT "x-Div."  ;  s  ' richiesta ingresso valore intervallo reticolo asse x

LOCATE 14, 66: INPUT "y-Div."  ;  t  ' richiesta ingresso valore intervallo reticolo asse y

LOCATE 15, 66: PRINT "-sn;+dx"     ' indicazione per l'istruzione successiva relativa allo spostamento
                                   ' a sinistra ( -sn) o a destra ( +dx) rispetto allo zero del tracciato
                                   ' della funzione

LOCATE 16, 66: INPUT "spost.x"  ;  g  ' richiesta ingresso entità di spostamento del tracciato della funzione

FOR  x = i  TO  f   STEP p  ' variazione della x nel campo stabilito a comando tra (inizio,i) e (fine,f)

y# = x ^ 4 + x ^ 3 + x ^ 2 + x + 1  ' funzione da studiare espressa a doppia precisione

r# = ( 16 / t ) * y#   ' calcolo variabile r a doppia precisione per ordinata istruzione PSET
                       ' comprensiva valore di scala stabilito a comando per l'asse y ( lettera t)

IF r# > 160 THEN r# = 160  ' istruzione che evita l'overflow positivo

IF r# < -160 THEN r# = -160  ' istruzione che evita l'overflow negativo        &
```

```
PSET ( 230 - ( 23 / s ) * ( -g ) + ( 23 / s ) * x , 160 - r# ), 15  ' genera il tracciato del grafico della funzione
                                                            ' in base al valore di scala stabilito
                                                            ' a comando per l'asse x (lettera s) , inoltre sposta il tracciato in base
                                                            ' al valore assegnato a comando esterno da "spost.x" tramite la (lettera g)

NEXT x  ' rimanda all'istruzione FOR x = i .....
```

F5

```
x inizio? -10
x fine?  +10
step.? .01
x-Div:? 1
y-Div.? 1
-sn ; +dx
spost. x ?  0
```

si ottiene il tracciato di figura 26 in cui è visibile un punto notevole (minimo della funzione)

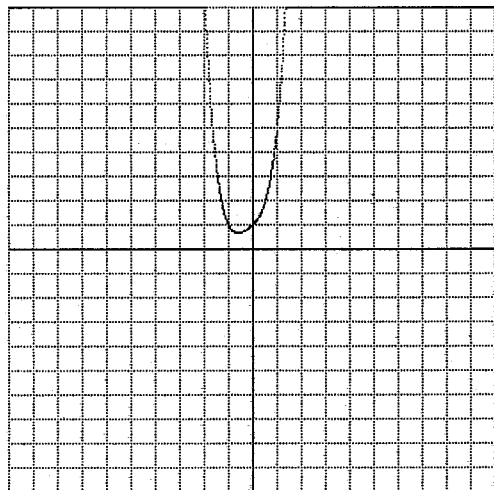



Figura 26
Grafico per ricerca punti notevoli funzione di quarto grado
Campo do variabilità della x :
da -10 a + 10
Scala asse x = 1 / div.
Scala asse y = 1 / div.

Dal tracciato si osserva:

-la funzione è in fuga nel campo dei valori positivi di y per x < -2 e x > +1.3
-nell'intervallo delle ascisse compreso tra -2 e +1.3 è presente una sella della funzione che ha il suo punto più basso per un valore di x valutabile con approssimazione in circa - .7; questo è l'unico punto notevole della funzione (punto di minimo).

Per determinare con maggior precisione l'ascissa del punto di minimo è ora necessario ricorrere alla computazione numerica della funzione in un piccolo intorno dell'ascissa del punto rilevato dal grafico.
L'operazione di calcolo si esegue, in base alle necessità, o a bassa precisione o ad elevata precisione:

Nel primo caso si sceglie un intervallo di variabilità della x tale da contenere il valore dell'ascissa del punto notevole stimata sul grafico e si esegue la computazione accettando i risultati che ne conseguono.
Nel secondo caso, visti i risultati del calcolo precedente, si ripete l'operazione per un intervallo della x più piccolo del primo a cavallo della ascissa del punto evidenziata con la prima computazione.
Il nostro esercizio viene svolto, a scopo didattico, su due intervalli:
Il primo intervallo viene fissato attorno al valore rilevato di -.7: da -.8 a -.4 e si esegue il computo mediante il seguente programma

```
CLS ' pulisce lo schermo

FOR x = -.8 TO -.4 STEP .1 ' campo di variabilità di x per il calcolo di 4 punti di y

Y = x ^ 4 + x ^ 3 + x ^ 2 + x + 1 ' funzione da computare

PRINT "x=" ; x ; "y=" ; y ' comando di stampa valori di x affiancati ai corrispondenti valori di y

NEXT x ' comando di ritorno a FOR x= ...
```

F5

```
x = -. 8   y = . 7376
x = -. 7   y = . 6871
x = -. 6   y = . 6736
x = -. 5   y = . 6875
```

F5

il calcolo mostra che il minimo della funzione è più vicino a x = -.6 che non al valore stimato di x = -.7

Se ripetiamo il procedimento di calcolo per un campo più stretto e centrato del primo, con un incremento della x pari ad 1 / 10 del precedente, abbiamo:

```
CLS ' pulisce lo schermo

FOR x = - .65 TO - .55 STEP .01 ' campo di variabilità di x per il calcolo dei punti di y

Y = x ^ 4 + x ^ 3 + x ^ 2 + x + 1 ' funzione da computare

PRINT "x=" ; x ; "y=" ; y ' comando di stampa valori di x affiancati ai corrispondenti valori di y

NEXT x ' comando di ritorno a FOR x= ...
```

F5

tra i valori presentati risulta che l'ordinata più piccola è y = .673577 alla quale corrisponde l'ascissa x = -.61, questa coppia risolve il nostro problema; la funzione ha un minimo di coordinate x = -.61 y = +.673

5.3 Esercitazione grafico-numerica n°2

La funzione che proponiamo per la ricerca dei punti notevoli, diversamente dalla precedente, sarebbe molto laboriosa da derivare con impossibilità pratica di risolvere le equazioni trascendenti ottenute dalle derivate stesse; in questo caso il metodo grafico numerico adottato è l'unico mezzo disponibile per ottenere buoni risultati in tempi estremamente brevi.

La funzione oggetto della nostra ricerca è

$$y = \frac{x^2 - 1}{x^2 + 1} e^{x/10}$$

che trasformata in simbologia Qbasic diventa

y = ((x ^ 2 - 1) / (x ^ 2 + 1)) * EXP (x / 10)

con essa completiamo il programma grafico:

```
LINE ( 230, 0 )-( 230, 320 )  ' ASSE  Y 4 quadranti scelti per visualizzare qualsiasi tipo di funzione

LINE ( 0, 160 )-( 460, 160 )  ' ASSE  X 4 quadranti scelti per visualizzare qualsiasi tipo di funzione

LOCATE 10, 66: INPUT "x inizio"  ; i   ' richiesta ingresso estremo inferiore campo variabilità di x

LOCATE 11, 66: INPUT "x fine"    ; f   ' richiesta ingresso estremo superiore campo variabilità di x

LOCATE 12, 66: INPUT "step"      ; p   ' richiesta ingresso valore dell'incremento di x (STEP)

LOCATE 13, 66: INPUT "x-Div."    ; s   ' richiesta ingresso valore intervallo reticolo asse x

LOCATE 14, 66: INPUT "y-Div."    ; t   ' richiesta ingresso valore intervallo reticolo asse y

LOCATE 15, 66: PRINT "-sn;+dx"         ' indicazione per l'istruzione successiva relativa allo spostamento
                                       ' a sinistra ( -sn) o a destra ( +dx) rispetto allo zero del tracciato
                                       ' della funzione

LOCATE 16, 66: INPUT "spost.x"   ; g   ' richiesta ingresso entità di spostamento del tracciato della funzione

FOR x = i TO f STEP p  ' variazione della x nel campo stabilito a comando tra (inizio,i) e (fine,f)

y# = ( ( x ^ 2 - 1 )/ ( x ^ 2 + 1 ) ) * EXP ( x / 10 )  ' funzione da studiare espressa a doppia
                                                        ' precisione

r# = ( 16 / t ) * y#  ' calcolo variabile r a doppia precisione per ordinata istruzione PSET
                      ' comprensiva valore di scala stabilito a comando per l'asse y ( lettera t)

IF r# > 160 THEN r# = 160  ' istruzione che evita l'overflow positivo

IF r# < -160 THEN r# = -160  ' istruzione che evita l'overflow negativo

PSET ( 230 - ( 23 / s ) * ( -g ) + ( 23 / s ) * x , 160 - r# ), 15  ' genera il tracciato del grafico
                                    ' della funzione in base al valore di scala
                                    ' stabilito a comando per l'asse x (lettera s) , inoltre sposta il tracciato in base
                                    ' al valore assegnato a comando esterno da "spost.x" tramite la (lettera g)

NEXT x  ' rimanda all'istruzione FOR x = i .....
```

dopo alcuni tentativi si trovano i valori più adatti per la migliore presentazione della funzione

x inizio? -10
x fine? +10
step.? .01
x-Div:? 1
y-Div.? .2
-sn ; +dx
spost. x ? 0

ottenendo il grafico di figura 27, dal grafico si osserva:

-la funzione è in fuga nel campo positivo delle y per x > 7.2
-la funzione tende a zero per x < -10
-nell'intervallo delle ascisse compreso tra - 4 e +1 sono evidenti due punti notevoli
-si ha un punto di minimo nell'intorno di x = 0
-si ha un punto di massimo relativo per x = -3.5 circa

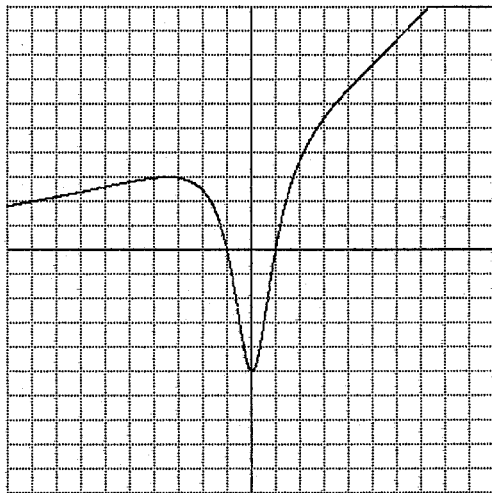



Figura 27
Grafico ricerca punti notevoli
funzione trascendente
Campo di variabilità della x :
da -10 a +10
Scala asse x = 1 / div
Scala asse y = .2 / div.

Per determinare l'ascissa del punto di minimo è ora necessario ricorrere alla computazione numerica della funzione in un piccolo intorno del punto rilevato dal grafico.
L'operazione di calcolo si esegue una sola volta, a bassa precisione, dato che la curva mostra con buona approssimazione la posizione del suo minimo.
L'intervallo viene fissato attorno al valore rilevato di 0: da -.5 a +.5 con incremento pari a .1.

```
CLS ' pulisce lo schermo

FOR x = -.5 TO .5 STEP .1 ' campo di variabilità di x per il calcolo di 10 punti di y

Y = ((x ^ 2 - 1) / (x ^ 2 + 1)) * EXP(x / 10) ' funzione da computare          &
```

PRINT "x=" ; x ; "y=" ; y ' comando di stampa valori di x affiancati ai corrispondenti valori di y

NEXT x ' comando di ritorno a FOR x= ...

F5

dalla serie di coppie x ed y presentate si stabilisce che il minimo della funzione ha, con buona approssimazione, le coordinate x = 0 y = -1.

F5

Per determinare l'ascissa del punto di massimo si ricorre alla computazione numerica della funzione in un piccolo intorno dell'ascissa del punto rilevato dal grafico.
L'intervallo viene fissato attorno al valore rilevato di - 3.5: da - 4 a - 3 e si esegue il computo mediante il programma

CLS ' pulisce lo schermo

FOR x = - 4 TO - 3 STEP .1 ' campo di variabilità di x per il calcolo dei punti di y

Y = ((x ^ 2 - 1) / (x ^ 2 + 1)) * EXP (x / 10) ' funzione da computare

PRINT "x=" ; x ; "y=" ; y ' comando di stampa valori di x affiancati ai corrispondenti valori di y

NEXT x ' comando di ritorno a FOR x= ...

F5

dalla serie di coppie x ed y presentate si stabilisce che il massimo relativo della funzione ha le coordinate

x = -3.4 y = +.5984311

CAPITOLO 6
DERIVATE DI FUNZIONI

L'argomento sviluppato in questo capitolo è relativo al calcolo ed alla presentazione grafica delle derivate di funzioni; si imposta e si risolve, mediate Qbasic, la forma analitica delle derivate sulla base della semplice definizione del rapporto incrementale.

6.1 La definizione di derivata di una funzione

E' noto che viene definita derivata di una funzione y = f(x) il limite del rapporto incrementale

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

il calcolo del limite è applicabile, in via di principio, a qualsiasi funzione f(x) che abbia caratteristiche di continuità; in pratica l'espressione è utilizzata prevalentemente nei libri di testo per dimostrazioni di principio che conducono a regole di calcolo applicative che di fatto non implicano lo sviluppo del limite sopra indicato.
Elenchi di derivate delle funzioni più comuni e di formule di derivazione sono disponibili come supporto per agevolare le operazioni di calcolo infinitesimale. Non sempre però il calcolo della derivata di una funzione è facilmente affrontabile con i mezzi matematici a disposizione, in alcuni casi, con funzioni complicate, l'operazione è difficoltosa.
Con l'ausilio del Qbasic si può determinare, numericamente, la derivata di qualsiasi funzione con estrema semplicità; questo tipo di elaborazione è illustrato nel paragrafo seguente.

6.2 La routine per il calcolo della derivata di una funzione

Il procedimento per il calcolo della derivata di una funzione, implementabile in Qbasic, si basa sulla definizione di rapporto incrementale riportata nel paragrafo 6.1.
L'operazione al limite viene sostituita, con buona approssimazione, da una serie di rapporti per valori di h molto piccoli, computati in un intervallo stabilito della variabile indipendente.
Si ha perciò che il calcolo del rapporto:

$$f'(x) \cong \frac{f(x+h) - f(x)}{h}$$

eseguito per n valori di x compresi nell'intervallo di variabilità da x1 a x2, assumendo h = 1 / 1000000, fornisce l'andamento approssimato della funzione derivata.
Il rapporto incrementale per il computo della derivata di una generica funzione viene implementato in Qbasic in doppia precisione come segue:

```
LINE ( 0, 160 )-( 460, 160 ) ' ASSI per 4 quadranti

LINE ( 230, 0 )-( 230, 320 )

LOCATE 15, 66: INPUT "x1=" ; x1  ' richiesta ingresso estremo inferiore campo di variabilità

LOCATE 16, 66: INPUT "x2=" ; x2  ' richiesta ingresso estremo superiore campo di variabilità          &
```

```
FOR  x = x1  TO  x2  STEP  ( x2 - x1 ) / 1000 ' definizione campo ed incremento di calcolo

h = 1 / 1000000   ' si fissa il valore di h per il calcolo del rapporto incrementale

yo# = f( x )  ' funzione originale in (x)

yi# = f( x + h )  ' funzione incrementata in (x+h)

yr# = ( yi# - yo# ) / h  ' calcolo del rapporto incrementale ( derivata di f(x) )

PSET ( 230 + k * x , 160 - k2 * yr# )  ' presentazione grafico f '(x)

NEXT x ' rimanda all'istruzione FOR x = per il calcolo dei 1000 punti di f '(x)
```

6.3 Esercitazioni elementari di derivazione e precisione di calcolo

Come si è osservato il programma compilato nel paragrafo 6.2 è molto semplice, i risultati che si ottengono sono però interessanti.
Proponiamoci ad esempio il calcolo della derivata della funzione y = Sen x anche se tale operazione non è necessaria essendo scontato che y' = Cos x
-si fissa il campo di variabilità della x da x1 = - 3.14 a x2 = 3.14
-si calcola il valore di k per l'istruzione PSET k = 230 / 3.14 = 73.25
-si calcola il valore di k2 per l'istruzione PSET k2 = 160 / 1 = 160

si compila il programma

```
LINE ( 0, 160 )-( 460, 160 ) ' ASSI 4 quadranti

LINE ( 230, 0 )-( 230, 320 )

LOCATE 15, 66: INPUT "x1=" ;  x1  ' richiesta ingresso estremo inferiore campo di variabilità

LOCATE 16, 66: INPUT "x2=" ;  x2  ' richiesta ingresso estremo superiore campo di variabilità

FOR  x = x1  TO  x2  STEP  ( x2 - x1 ) / 1000 ' definizione campo ed incremento di calcolo

h = 1 / 1000000   ' si fissa il valore di h per il calcolo del rapporto incrementale

yo# = SIN ( x )  ' funzione originale in (x)

yi# = SIN ( x + h )  ' funzione incrementata in (x+h)

yr# = ( yi# - yo# ) / h  ' calcolo del rapporto incrementale ( derivata di f(x) )

PSET ( 230 + 73.25 * x , 160 - 160 * yr# )  ' presentazione grafico f '(x)

NEXT x ' rimanda all'istruzione FOR x = per il calcolo dei 1000 punti di f '(x)
```

```
F5
                    x1= ? -3.14
                    x2= ? 3.14
```

dopo l'introduzione di x2 si ha la presentazione video della funzione derivata così come illustrato in figura 28. La curva, a tratto bianco, mostra chiaramente che l'andamento della derivata è, come ci attendevamo, una funzione cosinusoidale, può restare però il dubbio che il profilo non segua esattamente la legge y' = Cos x a causa dell'approssimazione di calcolo insita nel programma.

La funzione da derivare che abbiamo scelto consente un efficace controllo del grafico ottenuto dalla routine di programma dato che il risultato deve essere la funzione y' = Cos x che è facilmente tracciabile assieme alla curva calcolata.

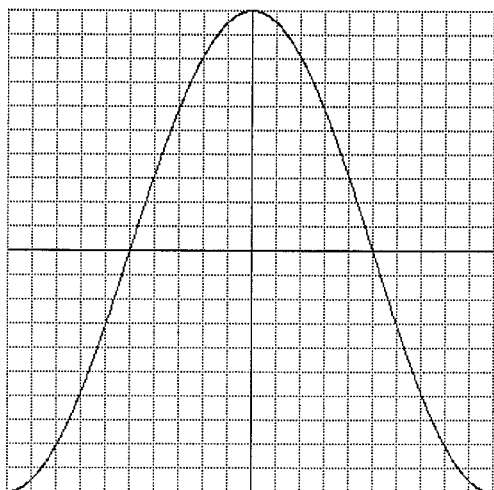



Figura 28
Grafico derivata funzione Sen x
nel campo x1 = -3.14 x2 = 3.14
Scala asse x = .314 rad./div.
Scala asse y = .1/div.

Se aggiungiamo al programma, dopo l'istruzione PSET esistente, una nuova istruzione PSET così strutturata:

PSET (230 + 73.25 * x , 160 - 160 * COS (x)),14 'presentazione grafico teorico della f '(x)

e ripetiamo l'esercizio, abbiamo il confronto grafico tra la f '(x) calcolata e la f '(x) teorica.

F5

x1= ? - 3.14
x2= ? 3.14

Dopo l'introduzione del valore di x2 si ha un nuovo tracciato in cui compare una sola curva cosinusoidale in giallo, ciò è la dimostrazione che il procedimento di calcolo sviluppato a programma conduce ad una funzione che è praticamente sovrapponibile a quella teorica con errori di approssimazione del tutto trascurabili.

Un secondo esempio per il controllo della precisione di calcolo è facilmente realizzabile mediante il calcolo della derivata di

$$y = e^x$$

che come è noto coincide con la funzione stessa $y' = e^x$:

-si fissa ad esempio il campo di variabilità della x da x1 = -1 a x2 = 1
-si calcola il valore di k per l'istruzione PSET k = 230 / 1 = 230
-si calcola il valore di k2 per l'istruzione PSET k2 = 160 / e = 58.86
compiliamo il programma:

```
LINE ( 0, 160 )-( 460, 160 ) ' ASSI per 4 quadranti

LINE ( 230, 0 )-( 230, 320 )

LOCATE 15, 66: INPUT "x1=" ; x1  ' richiesta ingresso estremo inferiore campo di variabilità

LOCATE 16, 66: INPUT "x2=" ; x2  ' richiesta ingresso estremo superiore campo di variabilità

FOR x = x1 TO x2 STEP ( x2 - x1 ) / 1000 ' definizione campo ed incremento di calcolo

h = 1 / 1000000 ' si fissa il valore di h per il calcolo del rapporto incrementale

yo# = EXP ( x ) ' funzione originale in (x)

yi# = EXP ( x + h ) ' funzione incrementata in (x+h)

yr# = ( yi# - yo# ) / h ' calcolo del rapporto incrementale ( derivata di f(x) )

PSET ( 230 + 230 * x , 160 - 58.86 * yr# ) ' presentazione grafico f '(x) calcolato

PSET ( 230 + 230 * x , 160 - 58.86 * yo# ),14 ' presentazione grafico f '(x) teorico

NEXT x ' rimanda all'istruzione FOR x = per il calcolo dei 1000 punti di f '(x)
```

F5

x1= ? - 1
x2= ? 1

dopo l'introduzione di x2 si ha la presentazione del grafico di figura 29 in cui è tracciata una sola curva, anche in questo caso il grafico della funzione derivata calcolata è perfettamente sovrapposto al grafico della derivata teorica.

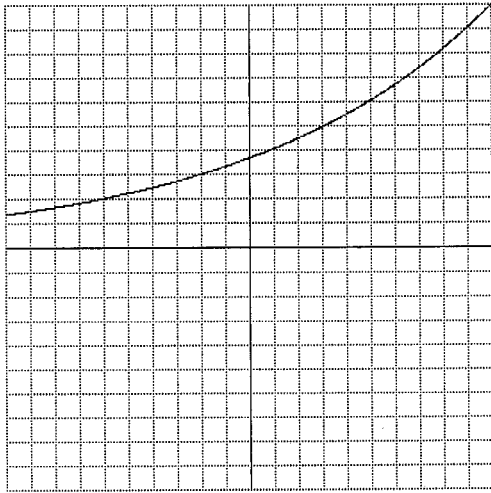



Figura 29
Grafico funzione derivata di $y = e^x$
comparato con y '
Scala asse x = .1 / div.
Scala asse y = .271 / div.

6.4 Esercizio generico di calcolo

Per chiudere questo capitolo è utile eseguire un ultimo esempio di computazione nel caso in cui la funzione da derivare abbia una espressione complicata, naturalmente questo esercizio non permette il controllo del risultato non conoscendo a priori la derivata teorica; si avrà la certezza del risultato in virtù di quanto è già stato dimostrato nel paragrafo precedente.

Sia data la funzione:

$$y = (1 / 3) \ln \frac{3 \, \text{Tang} (x / 2) - 2}{2 \, \text{Tang} (x / 2) + 3}$$

se ne voglia calcolare la derivata nell'intervallo compreso tra 1.5 e 3 radianti per lavorare in un tratto continuo e di validità della funzione; la corrispondenza simbolica della funzione in Qbasic è:

y = (1/3) * LOG((3 * TAN (x / 2) -2) / (2 * TAN (x / 2) + 3))

posto
x1 = 1.5 ; x2 = 3
per l'istruzione PSET si calcolano
k = 230 / 3 = 76.6 (per avere scala asse x = .3 / div.)
k2 = 160 / 2 = 80 (per avere scala asse y = .2 / div.)
si compila il programma

```
LINE ( 0, 160 )-( 460, 160 ) ' ASSI per 4 quadranti

LINE ( 230, 0 )-( 230, 320 )

LOCATE 15, 66: INPUT "x1=" ; x1 ' richiesta ingresso estremo inferiore campo di variabilità

LOCATE 16, 66: INPUT "x2=" ; x2 ' richiesta ingresso estremo superiore campo di variabilità

FOR x = x1 TO x2 STEP ( x2 - x1 ) / 1000 ' definizione campo ed incremento di calcolo

h = 1 / 1000000 ' si fissa il valore di h per il calcolo del rapporto incrementale

yo# = ( 1/3 ) * LOG( ( 3 * TAN ( x / 2 ) -2 ) / ( 2 * TAN ( x / 2 ) + 3 ) ) ' funzione originale in (x)

yi# = ( 1/3 ) * LOG( ( 3 * TAN ( ( x+h ) / 2 ) -2 ) / ( 2 * TAN ( ( x+h ) / 2 ) + 3 ) ) ' funzione incrementata

yr# = ( yi# - yo# ) / h ' calcolo del rapporto incrementale ( derivata di f(x) )

PSET ( 230 + 76.6 * x , 160 - 80 * yr# ) ' presentazione grafico f '(x) calcolato

NEXT x ' rimanda all'istruzione FOR x = per il calcolo dei 1000 punti di f '(x)
```

```
F5
                    x1= ? 1.5
                    x2= ? 3
```

dopo l'introduzione di x2 si ha la presentazione della funzione derivata così come è mostrato in figura 30 in cui è tracciata la curva che si sviluppa nel primo quadrante tra x1 = 1.5 ed il fondo scala x2 = 3.

Come abbiamo visto l'operazione di derivazione numerica e di presentazione grafica di una funzione è cosa molto agevole, si deve soltanto prestare attenzione nel fissare il campo di variabilità della x che deve assicurare alla f(x) sia la continuità che la validità.

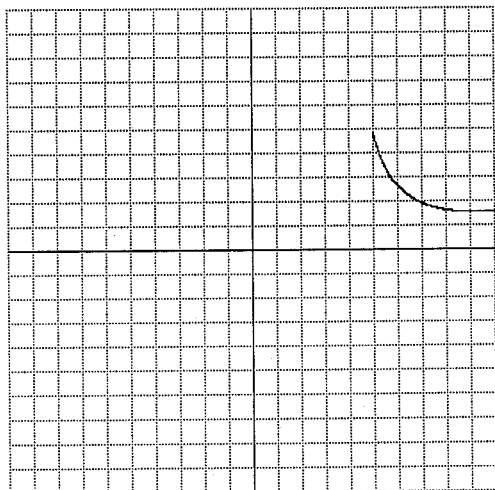



Figura 30
Grafico derivata della funzione data
Scala asse x = .3 / div.
Scala asse y = .2 / div.

CAPITOLO 7
INTEGRAZIONE DEFINITA DELLE FUNZIONI

Tratteremo in questo capitolo dell'integrazione definita delle funzioni, siano queste espresse mediante formule matematiche, siano esplicitate mediante tabelle. Vedremo l'utilità del Qbasic nel computo di questi integrali che sono di fondamentale importanza per numerose applicazioni in campo tecnico e didattico.

7.1 L'integrale definito di una funzione

Per gli sviluppi che andremo ad esporre nei paragrafi seguenti, giova ricordare i concetti basilari di analisi matematica relativi al calcolo dell'integrale definito di una funzione:
viene detto **integrale definito di $y = f(x)$** l'espressione:

$$I = \int_a^b f(x)\,dx$$

Il significato di questa simbologia è duplice
-il valore I rappresenta la sommatoria di infiniti prodotti tra la funzione $f(x)$ e l'incremento infinitesimale dx calcolati tra i limiti di variabilità di x compresi tra (a) e (b).
-il valore I che si ottiene dal calcolo dell'integrale rappresenta l'area della superficie sottesa dal tratto di curva $f(x)$, definita tra $x = a$ ed $x = b$.
Per entrambi i concetti il calcolo dell'integrale definito segue le stesse regole dettate dall'analisi matematica alle quali si rimanda il lettore interessato a rivedere l'argomento.
Per i nostri scopi non è necessario applicare le regole menzionate dato che affidiamo ad appositi programmi in Qbasic il computo dell'integrale in oggetto.

7.2 Il calcolo di un integrale definito

Prima di passare alla compilazione dei programmi applicativi per il computo dell'integrale definito è utile proporre, in termini estremamente semplici, un esempio di calcolo per un integrale elementare.
Sia data la funzione

$$y = x^2$$

se ne calcoli l'integrale definito nel campo di variabilità di x compreso tra $a = 0$ e $b = 10$ cioè:

$$I = \int_0^{10} (x)^2\,dx = \left[(x)^3 / 3 \right]_0^{10} = 333.333$$

il valore di I, ottenuto mediante il procedimento di analisi matematica, servirà da confronto con il primo risultato che otterremo applicando un programma in Qbasic per il computo di questo integrale.

7.3 Criterio di approssimazione per il calcolo dell'integrale definito

Nel paragrafo 7.1 abbiamo dato la definizione:

-il valore I rappresenta la sommatoria di infiniti prodotti tra la funzione f(x) e l'incremento infinitesimale dx calcolati tra i limiti di variabilità di x compresi tra (a) e (b),
questa definizione permette di ricavare una formula approssimativa per il calcolo dell'integrale definito implementabile in Qbasic; infatti l'espressione di sommatoria

$$I \cong \sum_{x=a}^{x=b} f(x)\, s$$

risponde alla definizione data per l'integrale salvo il numero finito dei prodotti e la sostituzione dell'infinitesimo dx con un numero s molto piccolo. I prodotti che entrano nella sommatoria, nell'assunto che x di f(x) si incrementi dello stesso valore s, hanno la seguente struttura:

$$f(a)\,s + f(a+s)\,s + f(a+2s)\,s + f(a+3s)\,s + \ldots\ldots\ldots + f(a+ns)\,s + f(b)\,s$$

Si comprende che più sono i prodotti e più è piccolo il valore di s tanto più la sommatoria approssimerà il valore teorico calcolato tramite l'integrale.

7.4 Il programma in Qbasic per il computo dell'integrale definito

Con le informazioni acquisite nei paragrafi precedenti procediamo alla stesura di un programma in Qbasic in grado di implementare la sommatoria:

$$I \cong \sum_{x=a}^{x=b} f(x)\, s$$

Ricordando quanto spiegato nel paragrafo 2.26 abbiamo tutti gli elementi per ottenere quanto vogliamo; riportiamo pertanto il programma di detto paragrafo adattandolo ai simboli della nuova sommatoria

```
CLS ' pulisce lo schermo

INPUT " a= " ; a ' richiesta di ingresso valore inferiore del campo di variabilità della x

INPUT " b= " ; b ' richiesta di ingresso valore superiore del campo di variabilità della x

INPUT " s=" ; s ' richiesta di ingresso valore sostituto dell'infinitesimo ed incremento di x

FOR x = a TO b STEP s ' impone che la variabile indipendente (x) inizi il calcolo assumendo il
                      ' valore (a) , quando l'esecuzione del programma giunge alla istruzione
                      ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
                      ' valore (s) . Il ciclo si ripete con incrementi di (s) per arrestarsi
                      ' quando il valore di x ha raggiunto il valore (b)

Y = f ( x ) ' funzione da integrare

P = s * Y ' prodotto della funzione per il sostituto s dell'infinitesimo

I = P + I ' somma al valore di I in memoria ( uguale a zero all'inizio) il primo valore di P , e
          ' sostituisce in memoria il valore di I=0 con il valore di I=P , dopo la prima istruzione
          ' NEXT, x si incrementa d i ( s ) e di conseguenza si ha un cambiamento di P; il valore     &
```

```
' di I in memoria si aggiorna sommandosi al nuovo valore di P, le somme si ripetono
' con ulteriori aggiornamenti del valore di I in memoria fino a quando x raggiunge il
' valore (b)  dell'estremo superiore del campo di variabilità

NEXT x  ' rimanda all'istruzione FOR per un successivo incremento di x

PRINT " I = " ;  I  ' comanda la presentazione del valore finale di  I  , corrispondente al calcolo
                    ' approssimato dell'integrale definito di f(x)
```

L'impiego di questo programma è mostrato nel paragrafo seguente .

7.5 Il calcolo di un integrale definito mediante programma in Qbasic

Applichiamo il programma compilato nel paragrafo 7.4 per il calcolo approssimato dell'integrale definito relativo alla funzione integrata analiticamente nel paragrafo 7.2; in questo modo si potranno confrontare i risultati dei due diversi sistemi di calcolo:
La funzione da integrare è

$$y = x^2$$

nell'intervallo della x compreso tra $a = 0$ e $b = 10$.
Si deve ricordare che dall'entità dell'incremento (s) dipende la bontà dell'approssimazione della sommatoria all'integrale definito; più è piccolo (s) minore è l'errore di calcolo che il programma commette, naturalmente valori molto piccoli di (s) richiedono tempi di elaborazione di macchina sensibili e ricorso a calcoli in doppia precisione. Vedremo che nel nostro esempio, con valori di s =.001, i risultati del calcolo a precisione singola sono molto buoni; ciò dipende dal numero dei prodotti che concorrono alla sommatoria. Infatti con il campo di variabilità di x tra 0 e 10 ed s =.001 si hanno (10 - 0) / .001 = 10000 punti di calcolo.
Con questi elementi completiamo il programma citato:

```
CLS ' pulisce lo schermo

INPUT " a= "  ;  a  ' richiesta di ingresso valore inferiore del campo di variabilità della x

INPUT " b= " ;   b  ' richiesta di ingresso valore superiore del campo di variabilità della x

INPUT " s= " ;    s   ' richiesta di ingresso valore sostituto dell'infinitesimo ed  incremento di x

FOR  x = a   TO  b  STEP  s   ' impone che la variabile indipendente (x) inizi il calcolo assumendo il
                              '  valore (a) , quando l'esecuzione del programma giunge alla istruzione
                              ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
                              '  valore (s) . Il ciclo si ripete con incrementi di (s) per arrestarsi
                              ' quando il valore di x ha raggiunto il valore (b)
Y =  x ^ 2   ' funzione da integrare

P = s  *  Y    ' prodotto della funzione per il sostituto  s  dell'infinitesimo

I = P + I   ' somma al valore di  I  in memoria  ( uguale a zero all'inizio) il primo valore di P , e
            ' sostituisce in memoria il valore di I=0  con il valore di I=P , dopo la prima istruzione
            ' NEXT,  x si incrementa d i ( s ) e di conseguenza si ha un cambiamento di P; il valore
            ' di I in memoria si aggiorna sommandosi al nuovo valore di P, le somme si ripetono
            ' con ulteriori aggiornamenti del valore di I in memoria fino a quando x raggiunge il
            ' valore (b)  dell'estremo superiore del campo di variabilità

NEXT  x  ' rimanda all'istruzione FOR per un successivo incremento di x
```

&

PRINT " I = " ; I ' comanda la presentazione del valore di I

```
                F5
a= ? 0
b= ? 10
s= ? .001
I = 333.270
                F5
```

Se confrontiamo il valore di I ottenuto dal programma I = 333.270 con il risultato ottenuto in precedenza per via analitica (I = 333.333) riscontriamo un errore per difetto pari a 333.333 / 333.270 = 1.000189 dell'ordine di circa 2 su 10000 che è irrilevante.
Se si ripete il computo con s =.01 (pari a 1000 punti di calcolo) si riscontra un errore dell' 1.5‰.
Se si sviluppa il programma con s = .1 (pari a 100 punti di calcolo) si riscontra un errore dell' 1.5%.
Queste osservazioni devono suggerire al lettore di adattare il valore di (s) alle dimensioni del campo di variabilità della x sulla base dei risultati che si vogliono ottenere.
La computazione dell'integrale definito di una funzione data può essere ripetuta per quante coppie del campo di variabilità della x sia necessario; i risultati si hanno immediatamente dalla routine di programma; se ad esempio cambiamo il campo del calcolo precedente in a = 7; b = 9 abbiamo I = 128.652.

7.6 La potenza del programma di calcolo dell'integrale definito

L'esercizio svolto nel paragrafo 7.5 è servito, sia per iniziare questa nuova procedura di calcolo, sia per avere un'idea della precisione che il metodo è in grado di offrire. Per fare ciò ci siamo serviti di una funzione molto semplice che siamo riusciti ad integrare anche per via analitica; in generale però l'integrazione analitica presenta notevoli difficoltà anche per chi ha confidenza con l'analisi matematica. Alcune funzioni non sono facilmente integrabili mentre il programma che abbiamo sviluppato offre, con attenzioni da porre sul tipo di funzione da trattare, una potenza di calcolo sorprendente per un mezzo tanto semplice.

7.7 Applicazione del programma per il calcolo di un integrale complicato

Vogliamo dimostrare l'efficacia del programma di calcolo mediante la soluzione di un integrale definito di una funzione complicata, integrale che pone un impegnativo problema di calcolo se affrontato con i metodi dell'analisi matematica.
L'integrale in oggetto è:

$$I = \int_{e}^{e^2} \frac{dx}{x(\ln^2 x + \ln^3 x)}$$

il valore di I, determinato analiticamente dopo lungo ed impegnativo lavoro di elaborazione matematica, è: I = .5 + ln .75 = .2123179
Lo stesso risultato si ottiene, in pochi secondi, mediante l'applicazione del programma di calcolo nel quale andremo ad inserire la funzione da integrare:

$$y = \frac{1}{x(\ln^2 x + \ln^3 x)}$$

trasformata mediante le corrispondenze simboliche in Qbasic:

y = 1 / (x * ((LOG (x)) ^ 2 + (LOG (x) ^ 3)))

La funzione deve essere integrata tra (e = 2.71828) ed (e^2 = 7.389) con un incremento (s) tale da ottenere circa 4000 punti di calcolo:

s = .001 < (7.389 - 2.71828) / 4000

Completiamo il programma:

```
CLS ' pulisce lo schermo

INPUT " a= " ; a  ' richiesta di ingresso valore inferiore del campo di variabilità della x

INPUT " b= " ; b  ' richiesta di ingresso valore superiore del campo di variabilità della x

INPUT " s= " ; s  ' richiesta di ingresso valore dell'incremento di x

FOR x = a  TO  b  STEP  s  ' impone che la variabile indipendente (x) inizi il calcolo assumendo il
                           ' valore (a) , quando l'esecuzione del programma giunge alla istruzione
                           ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
                           ' valore (s) . Il ciclo si ripete con incrementi di (s) per arrestarsi
                           ' quando il valore di x ha raggiunto il valore (b)

Y = 1 / ( x * ( ( LOG ( x ) ) ^ 2 + ( LOG ( x ) ^ 3 ) ) )  ' funzione da integrare

P = s * Y  ' prodotto della funzione per l'incremento (s) di  x

I = P + I  ' somma al valore di I in memoria ( uguale a zero all'inizio) il primo valore di P , e
           ' sostituisce in memoria il valore di I=0 con il valore di I=P , dopo la prima istruzione
           ' NEXT, x si incrementa d i ( s ) e di conseguenza varia il prodotto P; il valore
           ' di I in memoria si aggiorna sommandosi al nuovo valore di P, le somme si ripetono
           ' con ulteriori aggiornamenti del valore di I in memoria fino a quando x raggiunge il
           ' valore (b) dell'estremo del campo di variabilità

NEXT x  ' rimanda all'istruzione FOR per un successivo incremento di x

PRINT " I = " ; I  ' comanda la presentazione del valore di I
```

F5

```
a= ? 2.71828
b= ? 7.389
s= ? .001
I = .21243
```

NB. Data la complessità della funzione da integrare ed il numero elevato dei punti di calcolo il risultato si ottiene dopo un sensibile tempo di elaborazione macchina.

F5

Se confrontiamo il valore di I ottenuto dal programma I = .21243 con il risultato ottenuto per via analitica (I = .2123179) riscontriamo un errore per eccesso pari a 0.5 ‰ , scarto generalmente più che accettabile .

7.8 Il programma di calcolo per il controllo dell'integrale indefinito

Il programma per il computo degli integrali definiti può essere utilizzato con profitto per il controllo dei risultati relativi ad integrali indefiniti di funzioni.
A volte, dopo un lungo lavoro a tavolino, si ottiene come risultato di un integrale indefinito di una funzione una espressione della quale risulta molto difficoltoso controllare l'esattezza formale; con l'ausilio del programma di calcolo è possibile effettuare un controllo indiretto della validità dell'integrazione eseguita.
Se ad esempio si suppone di aver calcolato l'integrale indefinito della funzione

$$y = \text{Arctang}\,(x)^{1/2} \qquad (\text{funzione integranda})$$

ottenendo il seguente risultato:

$$I = \int \text{Arctang}\,(x)^{1/2}\,dx = (x+1)\,\text{Arctang}\,(x)^{1/2} - (x)^{1/2} + C \quad (\text{funzione integrale})$$

possiamo verificare la correttezza dello sviluppo computando, in un intervallo scelto a piacere, il valore del corrispondente integrale definito, sia manualmente con la funzione integrale, sia con il programma di calcolo con la funzione integranda; se i due metodi di calcolo porteranno a risultati praticamente uguali potremo ritenere corretta la funzione integrale ottenuta.
Vediamo come procedere alla verifica:

stabiliamo un intervallo di integrazione con $a = 0$ e $b = 1$
se impieghiamo la funzione integrale per il calcolo dell'integrale definito tra 0 ed 1 abbiamo:
$I = (2\,\text{Arctang}\,1 - 1) - (\text{Arctang}\,0 - 0) = .5707$
se impieghiamo il programma di calcolo con la funzione integranda questa deve essere scritta in simbologia Qbasic:

y = ATN (SQR (x))

dato poi che l'intervallo di integrazione è $(1 - 0) = 1$, per ottenere 10000 punti di calcolo si deve porre s = .0001, avremo perciò:

```
CLS ' pulisce lo schermo

INPUT " a= " ; a ' richiesta di ingresso valore inferiore del campo di variabilità della x

INPUT " b= " ; b ' richiesta di ingresso valore superiore del campo di variabilità della x

INPUT " s= " ; s ' richiesta di ingresso valore dell'incremento di x

FOR x = a TO b STEP s ' impone che la variabile indipendente (x) inizi il calcolo assumendo il
                      ' valore (a) , quando l'esecuzione del programma giunge alla istruzione
                      ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
                      ' valore (s) . Il ciclo si ripete con incrementi di (s) per arrestarsi
                      ' quando il valore di x ha raggiunto il valore (b)

Y = ATN ( SQR ( x ) ) ' funzione da integrare

P = s * Y ' prodotto della funzione per l'incremento (s) di x

I = P + I ' somma al valore di I in memoria ( uguale a zero all'inizio) il primo valore di P , e
          ' sostituisce in memoria il valore di I=0 con il valore di I=P , dopo la prima istruzione      &
```

```
' NEXT , x si incrementa d i ( s ) e di conseguenza cambia il prodotto P; il valore
' di I in memoria si aggiorna sommandosi al nuovo valore di P, le somme si ripetono
' con ulteriori aggiornamenti del valore di I in memoria fino a quando x raggiunge il
' valore (b) dell'estremo del campo di variabilità

NEXT x ' rimanda all'istruzione FOR per un successivo incremento di x

PRINT " I = " ; I ' comanda la presentazione del valore di I
```

F5

```
a= ? 0
b= ? 1
s= ? .0001
I = . 5707569
```

F5

Dai due diversi modi di computazione risultano valori di I molto vicini tra loro, ciò significa che il calcolo dell'integrale indefinito ha portato ad un risultato corretto.

7.9 Sulle discontinuità delle funzioni da integrare

Prima di chiudere questo argomento è indispensabile ricordare al lettore che alcune funzioni matematiche a carattere "discontinuo" non possono essere integrate senza alcuni accorgimenti che, escludendo le discontinuità, consentano al programma di lavorare correttamente.
Una funzione discontinua di seconda specie, ad esempio, presenta la discontinuità nell'intorno di valori positivi e/o negativi di x infinitamente piccoli, tali valori, se non opportunamente esclusi, portano il Qbasic a denunciare **Divisione per zero** e il programma di calcolo si blocca.
Se la funzione da integrare non è notoriamente continua nell'intervallo considerato è necessario eseguire un controllo, prima di procedere all'integrazione, per accertarsi delle sue caratteristiche e adottare alcune precauzioni.
Vediamo come operare se una funzione è discontinua (discontinuità di seconda specie); prendiamo in esame l'espressione:

$$y = \frac{1}{x - 1}$$

la discontinuità di questa funzione è evidente; quando x tende a 1 per $x < 1$ la funzione tende ad meno infinito, quando x tende a 1 per $x > 1$ la funzione tende a più infinito. Se la funzione deve essere integrata in un campo di variabilità che non contiene l'ascissa $x = 1$, per la quale nasce la discontinuità, il programma di calcolo può essere applicato normalmente. La funzione può essere integrata ad esempio tra 0 e .7; tra 1.5 e 4; ecc.. dato che in questi intervalli non è compreso il valore $x = 1$.
Se invece c'è l'esigenza di calcolare l'integrale tra 0 ed 1 è d'obbligo ridurre l'intervallo tra 0 e (1-i) in questo modo si evita la discontinuità e l'errore che si commette è tanto più piccolo quanto è piccolo i .
Se invece si deve calcolare l'integrale in un intervallo che contiene il punto di discontinuità, per esempio tra .5 e 1.5, si deve dividere l'intervallo in due parti, la prima tra .5 e (1 - i), la seconda tra (1 + i) e 1.5, per ciascun intervallo deve essere eseguito il computo a programma ed il valore complessivo dell'integrale sarà dato dalla somma dei due integrali parziali. Il computo sarà tanto più preciso quanto i sarà piccolo.

7.10 Integrazione delle funzioni di tabella

E' di notevole importanza pratica l'integrazione delle funzioni di tabella, questo calcolo dà modo di ricavare preziosi elementi di sintesi dalla matrice che altrimenti non sarebbero utilizzabili.
Nel capitolo 3, paragrafo 3.29, abbiamo trattato delle matrici di tipo permanente, qui invece tratteremo delle matrici di tipo volatile che meglio si prestano alle operazioni di integrazione secondo il programma di calcolo introdotto nel paragrafo 7.4.
L'impostazione di una matrice di tipo volatile richiede un semplice programma in Qbasic che andiamo di seguito a compilare e commentare:

```
DIM a ( 255 )  ' fissa il numero massimo degli elementi  che devono formane la matrice volatile

INPUT "M=" ;  M  ' chiede all'operatore quanti sono gli elementi della matrice

FOR x = 1 TO M  ' imposta la richiesta automatica degli M valori di a(x) con i quali si deve
                ' costruire la matrice volatile

PRINT "a" ;  x  ' stampa il simbolo dell'elemento di matrice con accanto il valore di x scandito
                ' dall'istruzione FOR , ad esempio  a 5 ,che indica il quinto elemento della matrice

INPUT a( x )  ' chiede all'operatore l'introduzione del valore dell'elemento di matrice individuato
              ' nell'istruzione precedente , ad esempio se a 5 = . 35 si ha :  a5    ? .35

NEXT x   ' rimanda all'istruzione FOR x = ... per la prosecuzione dell'introduzione di tutti gli M
         ' elementi della matrice

FOR  x = 1  TO  M  ' dopo l'introduzione di tutti gli elementi della matrice procede a richiamarli per
                   ' la loro utilizzazione sotto forma di funzione di matrice

y =  a ( x )  ' funzione di matrice da utilizzare per l'integrazione

NEXT x  ' rimanda alla seconda istruzione FOR x = ..per esplorare tutti gli elementi della matrice
```

Visto come organizzare una funzione di matrice è necessario trattare dell'integrale definito di questo tipo di funzioni che può seguire diverse regole di impostazione; la regola più semplice, che noi adotteremo, è detta dei trapezi; essa prevede il calcolo dell'integrale definito mediante la formula:

$$I = (.5\, a(1) + a(2) + a(3) + ... + a(M - 1) + .5\, a(M)) (f - i) / (M-1)$$

in cui a(1); a(2) ..; a(M) sono gli M elementi che compongono la matrice, i ed f sono rispettivamente l'inizio e la fine dell'intervallo entro il quale sono stati rilevati gli elementi stessi; un esempio chiarirà quanto detto:
Se consideriamo un arco della parabola $Y = X^2$ nell'intervallo della X compreso tra 1 e 6 e andiamo a rilevarne 6 valori per formare una tabella abbiamo:

X	Y
1	1
2	4
3	9
4	16
5	25
6	36

da questa tabella possiamo passare alla funzione di matrice soltanto cambiando i simboli delle variabili, al posto della variabile indipendente X si scrive la lettera x al posto della variabile indipendente Y si scrive il simbolo a(x).

x	a(x)	Questa è una funzione di matrice in cui:	
1	1	a(1) = 1	
2	4	a(2) = 4	
3	9	a(3) = 9	definita nell'intervallo 1; 6 in cui sono
4	16	a(4) = 16	stati ricavati gli elementi ;
5	25	a(5) = 25	sarà quindi inizio i = 1
6	36	a(6) = 36	fine f = 6

Data la formula per il calcolo dell'integrale definito e chiariti gli aspetti formali della funzione di matrice compiliamo e commentiamo la nuova struttura del programma necessario al computo dell'integrale stesso; il programma impiega l'istruzione condizionata **IF** in cascata mai utilizzata prima, questa si presenta nella forma:

IF x = 1 **THEN** k = .5 **ELSE IF** x = M **THEN** k = .5 **ELSE** k = 1

da cui il programma:

```
CLS  ' pulisce lo schermo

DIM a( 6 )  ' stabilisce il numero degli elementi della matrice volatile

INPUT "i=" ; i  ' chiede l'introduzione del valore iniziale dell'intervallo in cui sono stati ricavati gli
                ' elementi di matrice (i = inizio)

INPUT "f=" ; f  ' chiede l'introduzione del valore finale dell'intervallo (f = fine)

INPUT "M=" ; M  ' chiede il numero degli elementi che compongono la matrice

FOR x = 1 TO M  ' imposta la richiesta automatica degli M valori di a(x)con i quali si deve
                ' costruire la matrice volatile

PRINT "a" ; x  ' stampa il simbolo dell'elemento di matrice con accanto il valore di x scandito
               ' dall'istruzione FOR , ad esempio a 5 ,che indica il quinto elemento della matrice

INPUT a( x )  ' chiede all'operatore l'introduzione del valore dell'elemento di matrice individuato
              ' nell'istruzione precedente mediante la presentazione del simbolo ( ? )

NEXT x  ' rimanda all'istruzione FOR x =... per la prosecuzione dell'introduzione di tutti gli M
        ' elementi della matrice

FOR x = 1 TO M  ' dopo l'introduzione di tutti gli elementi della matrice procede a richiamarli per
                ' la loro utilizzazione sotto forma di funzione di matrice

IF x = 1 THEN k = .5 ELSE IF x = M THEN k = .5 ELSE k = 1
        ' istruzione che condiziona il primo e l'ultimo elemento della matrice,
        ' moltliplicandoli per k = . 5 così come prevede la formula dei trapezi

y = a( x )  ' funzione di matrice per il calcolo dell'integrale

P = k * y * ( f - i ) / ( M -1 )  ' calcolo del singolo addendo dell'integrale secondo la formula dei trapezi

E = P + E  ' somma al valore di E in memoria ( uguale a zero all'inizio) il primo valore di P , e
           ' sostituisce in memoria il valore di E=0 con il valore di E=P , dopo la prima istruzione      &
```

' NEXT x si ha l'introduzione a calcolo del secondo elemento di matrice di conseguenza
' cambia anche il valore di P, si ha un aggiornamento di E in memoria con la somma
' di E al nuovo valore di P, le somme si ripetono
' con ulteriori aggiornamenti del valore di E in memoria fino a quando x raggiunge
' il valore M relativo all'ultimo elemento della matrice

NEXT x ' rimanda alla seconda istruzione FOR per la scansione della matrice

PRINT "I =" ; E ' presenta il valore I = E dell'integrale definito della funzione matrice

E' molto interessante procedere ora al calcolo dell'integrale definito della funzione di tabella che abbiamo costruito con i dati ricavati dalla parabola:

```
                F5
i= ? 1
f= ? 6
M= ? 6
a1
? 1
a2
? 4
a3
? 9
a4
? 16
a5
? 25
a6
? 36
I = 72.5

                F5
```

Il computo ci ha fornito un valore dell'integrale definito pari a I = 72.5; dato che la funzione di matrice è stata ricavata dalla funzione parabola è significativo il confronto tra il valore di I, calcolato utilizzando soltanto sei punti della funzione originale, e l'integrale definito della funzione originale che, calcolato analiticamente, fornisce I = 71.666; come si vede l'errore commesso utilizzando la funzione matrice è dello 1.16 % in eccesso, risultato molto soddisfacente.
I valori che abbiamo inserito nella matrice, essendo quest'ultima di tipo volatile, si perdono naturalmente quando il programma viene salvato nella memoria del P.C. Dovranno pertanto essere reinseriti uguali o diversi valori di matrice per ulteriori impieghi della routine di calcolo.

CAPITOLO 8
GLI ALGORITMI DI FOURIER

Sono detti algoritmi di Fourier una serie di strumenti matematici che permettono di eseguire, sia per via puramente analitica, sia per via numerica, l'analisi dello spettro di frequenza che caratterizza i fenomeni dinamici. Questo capitolo è dedicato alla implementazione in Qbasic di alcuni tra i più utili ed importanti algoritmi di Fourier.

8.1 I fenomeni periodici

Vanno sotto il nome di fenomeni periodici quegli eventi fisici che si ripetono ad intervalli di tempo regolari. La nota pura di un diapason, ad esempio, è un fenomeno periodico rappresentato dalla funzione $y = \text{Sen}(2\pi f t)$ in cui f è la frequenza della vibrazione acustica e t è il tempo. In questo caso la natura del fenomeno è espressa completamente dalla funzione matematica che ne mette in evidenza, sia la natura sinusoidale, sia la frequenza del suono generato.
Nel caso invece del rombo di un motore a scoppio il suono è periodico a frequenza base f_0 ma non è puro, esso è la composizione di un insieme di vibrazioni che non possono essere rappresentate da una espressione sintetica e trasparente come la precedente. Per molti fenomeni periodici sono state studiate funzioni in grado di rappresentarne le caratteristiche dinamiche in dipendenza del tempo; queste funzioni non forniscono però informazioni sul tipo di componenti ondulatorie che ne costituiscono la struttura.
La serie di Fourier è lo strumento per analizzare la struttura dei fenomeni periodici al fine di stabilire quali sono le componenti ondulatorie che li caratterizzano.

8.2 La serie di Fourier per i fenomeni periodici

In questo paragrafo tratteremo della serie di Fourier al solo scopo di creare le premesse per le successive implementazioni degli algoritmi in Qbasic, l'argomento sarà perciò soltanto accennato senza alcuna dimostrazione che non potrebbe comunque essere svolta in un testo di questo tipo.
La serie di Fourier dà modo di trasformare una funzione periodica del tempo

$$y = f(2\pi f_0 t)$$

in cui f_0 è la frequenza del periodo base, in una sommatoria di infinite componenti trigonometriche che ne evidenziano il contenuto a carattere frequenziale (spettro di frequenza) mediante oscillazioni multiple di f_0. La serie è esplicitata come segue:

$$y = A_0 + \sum_{n=1}^{n=\infty} A_n \text{Cos}(n 2\pi f_0 t) + B_n \text{Sen}(n 2\pi f_0 t)$$

dove $A_0 = 1/2\pi \int_{-\pi}^{+\pi} f(2\pi f_0 t)\, d(2\pi f_0 t)$

$$A_n = 1/\pi \int_{-\pi}^{+\pi} f(2\pi f_0 t) \text{Cos}(n 2\pi f_0 t)\, d(2\pi f_0 t)$$

$$B_n = 1/\pi \int_{-\pi}^{+\pi} f(2\pi f_0 t)\, \mathrm{Sen}(n\, 2\pi f_0 t)\, d(2\pi f_0 t)$$

sono detti i coefficienti della serie.
Ciascun elemento della serie rappresenta in ampiezza e fase una delle frequenze dello spettro che compongono il fenomeno periodico.
L'ampiezza di ciascuna delle n frequenze (righe) che compongono lo spettro è data da:

$$C_n = (A_n^2 + B_n^2)^{1/2}$$

C_n è detto il modulo delle coppie $A_n \mathrm{Cos}(n\, 2\pi f_0 t) + B_n \mathrm{Sen}(n\, 2\pi f_0 t)$ che costituiscono la serie di Fourier.
L'ampiezza del valor medio del fenomeno coincide con A_0.
Il coefficiente A_0 viene calcolato una sola volta, i coefficienti A_n; B_n devono essere calcolati per tutti i valori di n corrispondenti al numero delle **frequenze significative** che compongono lo spettro.
Si è accennato alle frequenze significative dello spettro perché da un certo valore in poi la loro ampiezza, che scaturisce dal calcolo della serie, può diventare praticamente irrilevante.
Le formule possono essere ostiche a chi non ha dimestichezza in questo campo, ciò non deve preoccupare il lettore perché non sarà costretto ad impiegarle. Le formule sono mostrate soltanto per evidenziare un fatto fondamentale:

La serie di Fourier per i fenomeni periodici si ottiene da diverse elaborazioni della funzione data $y = f(2\pi f_0 t)$, mediante tre integrali per il calcolo di A_0, A_n, B_n.

Quindi per applicare la serie di Fourier deve essere espressa in forma analitica la funzione del fenomeno che si vuole analizzare; dato che sono disponibili elenchi completi di funzioni delle quali è già stata computata la serie in oggetto non è necessario applicare la procedura indicata se non per fare esercizio su questo argomento. A questo punto ci si chiederà perché aver discusso su questo tema se già tutto è stato fatto; la risposta è semplice, nel lavorare quotidiano si incontrano fenomeni periodici che non sono definiti mediante una funzione matematica, essi sono presentati ai nostri sensi, o con registrazioni grafiche, o con presentazioni oscilloscopiche, od infine con tracciati manuali a punti, in questi casi non si può procedere con sviluppi analitici per la ricerca delle componenti frequenziali, ma sulla base di quanto abbiamo mostrato si possono ottenere risultati soddisfacenti implementando opportunamente in Qbasic un adattamento della serie di Fourier.

8.3 Un esempio della serie di Fourier per i fenomeni periodici

Senza addentrarci in sviluppi analitici complicati mostriamo un esempio della serie di Fourier applicata ad un fenomeno periodico definito completamente mediante una funzione matematica. L'esempio è preso da un elenco di serie di Fourier già calcolate, esso si riferisce ad una funzione del tempo quale l'onda rettangolare simmetrica riportata in figura 31.
La funzione matematica che definisce questo fenomeno periodico è data da:

$$f(2\pi f_0 t) = \begin{cases} -E; & \text{per } (2k-1)\pi < 2\pi f_0 t < (2k\pi) \\ +E; & \text{per } (2k\pi) < 2\pi f_0 t < (2k+1)\pi \end{cases}$$

per $k = 0$; +/- 1; +/- 2; +/-3; .. ecc..

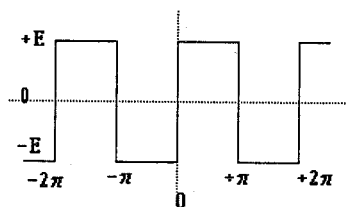



Figura 31
Onda rettangolare

Questo particolare algoritmo sintetizza il seguente ragionamento:

-la funzione ha ampiezza uguale a $-E$ per tutti i valori di $(2 \pi f_0 t)$ compresi tra $(2k-1)\pi$ e $(2k\pi)$.

-la funzione ha ampiezza uguale a $+E$ per tutti i valori di $(2 \pi f_0 t)$ compresi tra $(2k\pi)$ e $(2k+1)\pi$

quanto sopra per $k = 0$; +/- 1; +/- 2; +/-3; .. ecc..

Della funzione in oggetto è stata calcolata analiticamente la serie di Fourier che è sotto rappresentata

$$f(wt) = (4E/\pi)[\ (\text{Sen } wt) + (\text{Sen } 3wt)/3 + (\text{Sen } 5wt)/5 + (\text{Sen } 7wt)/7.. + ...]$$

dove $w = (2 \pi f_0)$

" si impiega il simbolo w in vece di ω dato che quest' ultimo non è disponibile in Qbasic"

La serie è già completamente sviluppata ed i singoli addendi rappresentano gli elementi frequenziali che compongono l'onda rettangolare; in questo caso si osserva:

-la serie è mancante del coefficiente A_0, ($A_0 = 0$), ciò indica che il valor medio dell'onda è nullo
-la serie è mancante di tutti i termini $A_n \text{Cos}(n 2 \pi f_0 t)$ si ha cioè per tutti i valori di n: ($A_n = 0$)
-la serie ha i coefficienti B_n per i soli valori dispari di n: $B1 = 4E/\pi$; $B3 = 4E/3\pi$; $B5 = 4E/5\pi$; $B7 = 4E/7\pi$
-il fenomeno periodico è formato da infinite frequenze multiple dispari della frequenza base f_0.
il primo addendo è un'onda sinusoidale alla frequenza base f_0, il secondo addendo è un'onda sinusoidale alla frequenza $3f_0$ con ampiezza pari ad 1/3 del primo, il terzo addendo è un'onda sinusoidale alla frequenza $5f_0$ con ampiezza pari ad 1/5 del primo, ecc.
-in questo caso il valore C_n dell'ampiezza delle righe coincide con B_n, per n dispari.
L'onda originale è ricostruibile con la serie data sommando tra loro gli infiniti addendi che la compongono. In pratica ciò non è possibile, la somma può essere estesa, a titolo di esercitazione, ad un numero elevato, ma finito, di addendi che consentono in effetti la ricostruzione approssimata dell'onda. Osservando però che i termini della serie decrescono rapidamente in ampiezza si comprende che l'errore di ricostruzione può essere molto contenuto anche con un numero finito di essi. Per provare quanto sopra vediamo come implementare un programma per la ricostruzione dell'onda rettangolare sulla base della serie di Fourier e dei dati che da essa abbiamo dedotto.

Il programma è strutturato per il controllo dei coefficienti A_o; A_n; B_n ricavati dall'applicazione della serie di Fourier ad una qualsiasi funzione periodica; con esso si esegue la ricostruzione della funzione ed il suo tracciamento grafico, operando un procedimento inverso a quello che ha condotto alla determinazione dello spettro frequenziale.
Infatti per la determinazione dello spettro della funzione da analizzare la si scompone mediante n addendi della serie di Fourier, per la ricomposizione della funzione analizzata si esegue la sommatoria di n addendi della serie di Fourier.
Compiliamo e commentiamo il programma di ricostruzione che si avvale di funzioni di matrici volatili per accogliere i valori dei coefficienti A_n; B_n.
Il programma impiega l'istruzione **DIM** per il dimensionamento di matrice in forma duplice; per dimensionare due matrici contemporaneamente si scrive: **DIM A(255),B(255)**.

```
LINE ( 230, 0 )-( 230, 320 ) ' ASSE Y 4 quadranti

LINE ( 0, 160 )-( 460, 160 ) ' ASSE X 4 quadranti

DIM A(255) , B(255) ' impostazione delle dimensioni di due matrici volatili per accogliere An ; Bn

LOCATE 10, 66: INPUT "n" ; k ' richiesta del numero n degli addendi con i quali si vuole realizzare
                             ' la sommatoria della serie di Fourier , tutte le istruzioni LOCATE sono
                             ' dimensionate per far comparire le richieste all'esterno del reticolo

LOCATE 11, 66: INPUT "Ao" ; Ao ' richiesta di introduzione del valore di Ao

FOR n = 1 TO k ' comando automatico per richiesta n valori di matrice per le due matrici

LOCATE 14, 66: PRINT "A" ; n ' stampa il simbolo An per la successiva istruzione

LOCATE 15, 66: INPUT A( n ) ' stampa il simbolo ? per l'introduzione del valore di A(n)
                            ' relativo alla prima matrice

LOCATE 18, 66: PRINT "B" ; n ' stampa il simbolo Bn per la successiva istruzione

LOCATE 19, 66: INPUT B( n ) ' stampa il simbolo ? per l'introduzione del valore di B(n)
                            ' relativo alla seconda matrice

NEXT n ' rimanda all'istruzione FOR n = ... per completare le due matrici

FOR wt = 0 TO 6.28 STEP .00628 ' fissa la variabilità dell'argomento dei termini trigonometrici
                               ' per le componenti Coseno e Seno della sommatoria della serie
                               ' con incremento di wt tale da ottenere 1000 punti di calcolo

C = Ao ' pone il valore di C uguale ad Ao per sommarlo alla serie

FOR n = 1 TO k ' comando automatico per la lettura dei valori delle due matrici A(n) ; B(n)

C = C + A( n ) * COS( n * wt ) + B( n ) * SIN( n * wt ) ' sommatoria progressiva in base ai prodotti
                                                        ' An Cos(n wt) e Bn Sen(n wt) secondo la serie
                                                        ' di Fourier

NEXT n ' rimanda all'istruzione FOR n= ... per la scansione delle due matrici

PSET ( 230 + 36 * wt , 160 - 160 * C ), 14 ' comanda il tracciamento del grafico della funzione
                                           ' ricostruita

NEXT wt ' rimanda all'istruzione FOR wt= per il calcolo di un successivo punto della funzione ricostruita
```

Ora non resta che provare il programma inserendo i dati emersi dall'analisi dell'onda rettangolare per una sommatoria della serie con un numero finito di termini, ad esempio con 20 addendi si ha:

per semplificare fissiamo il valore di $E = \pi / 4$ in questo modo abbiamo

Ao = 0
An = 0 per tutti i valori di n
Bn = 0 per tutti gli n pari
B1 = 1; B3 =.333; B5 =.2; B7 =.142; B9 =.111; B11 =.09; B13 =.076; B15 =.066;
B17 =.058; B19 =.052

F5

n ? 20
A0 ? 0
A1
? 0
B1
? 1

....

si procede, dopo cancellazione dei dati precedenti con i tasti freccia destra freccia sinistra, ad inserire:

....

tutti 0 per A2; A3; ...A20
tutti 0 per B2; B4; ...B20
i valori sopra elencati per B3 ;B5 ;...B19

dopo la ventesima coppia si ha la presentazione grafica della ricostruzione di un periodo dell'onda rettangolare come mostra la figura 32.

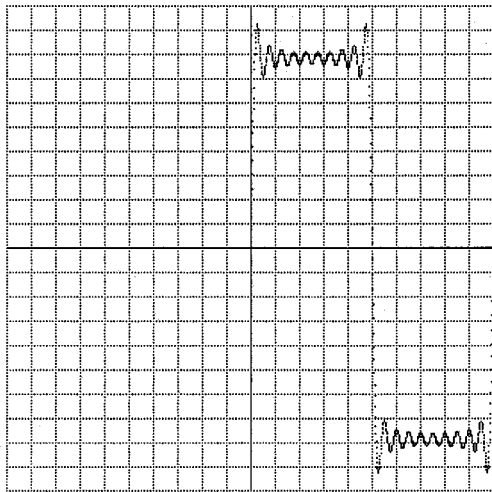



Figura 32
Periodo d'onda rettangolare ricostruito mediante serie di Fourier

Dalla figura si osserva che i fronti dell'onda riprodotta sono notevolmente ripidi e che approssimano abbastanza bene l'onda originale, nei tratti in cui l'onda originale è piatta, intervalli compresi tra 0 e 3.14 e tra 3.14 e 6.28, l'approssimazione è lasciata all'ondulazione della componente a frequenza più elevata relativa alla 19ª armonica che è l'ultimo termine, diverso da 0, dei 20 addendi messi a calcolo.
Dal buon risultato ottenuto possiamo affermare, ed in questo caso è ovvio, che i coefficienti della serie calcolata sono corretti.
Le informazioni ricavate da questo esempio mostrano la potenzialità dell'algoritmo e ci forniscono gli elementi, utili nel prosieguo del capitolo, per il controllo dei programmi che andremo ad implementare.

8.4 Approssimazione della serie di Fourier

Se il fenomeno periodico di cui vogliamo determinare le componenti frequenziali non è definito mediante una determinata legge matematica, ed è questo il caso più frequente in pratica, non è possibile svilupparlo mediante la serie di Fourier illustrata nel paragrafo 8.2. In questi casi si può procedere con diversi sistemi che presuppongono la disponibilità del tracciato del fenomeno, sotto forma di registrazione grafica od altro, dalla quale rilevare un certo numero di valori dell'ampiezza da utilizzare per i calcoli successivi.
Un sistema si avvale della serie di Fourier di cui al citato paragrafo, opportunamente modificata sì da ottenere una espressione approssimata della stessa come sotto indicato:

$$y \cong A'_0 + \sum_{n=1}^{n=m} A'_n \text{Cos}(n\, 2\pi f_0 t) + B'_n \text{Sen}(n\, 2\pi f_0 t)$$

in cui i coefficienti A'_0; A'_n; B'_n non vengono calcolati con procedimenti analitici ma con elaborazioni numeriche al computer. I tre integrali che definiscono teoricamente i coefficienti vengono sostituiti dal computo di tre integrali per funzioni di tabella quali quelli sviluppati nel paragrafo 7.10.

8.5 Implementazione della serie approssimata di Fourier

L'implementazione della serie approssimata di Fourier si basa sull'espressione del paragrafo precedente e sul calcolo dei coefficienti con la particolare modalità che abbiamo acquisito nel computo degli integrali di funzioni di tabella.
Per mostrare il procedimento di implementazione in Qbasic, che permette di determinare lo spettro di frequenza di un dato fenomeno periodico, è opportuno ricorrere ad un esempio pratico sulla base di un ipotetico rilievo oscillografico del fenomeno stesso:
sia dato pertanto il grafico di figura 33.

Il tracciato mostra un periodo di un'onda non simmetrica della quale ci proponiamo di determinare lo spettro di frequenza.
Si tratta ora di ricavare dal grafico dato una funzione di tabella che ci dia modo di procedere al calcolo dei coefficienti A'_0; A'_n; B'_n e delle ampiezze C'_n della serie di Fourier, mediante integrazione a mezzo Qbasic.
Per iniziare la procedura è necessario dividere il periodo dell'onda, pari a 2π, in un numero discreto di intervalli uguali così come è mostrato, ad esempio, in figura 34:

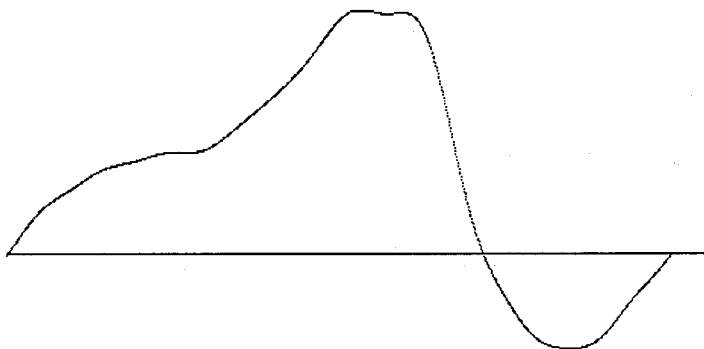



Figura 33
Onda periodica da analizzare
nel campo della frequenza

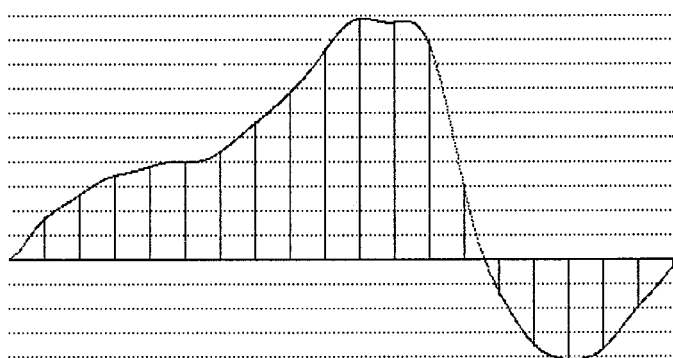



Figura 34
Onda periodica campionata per
analisi nel campo della frequenza

La pratica ha mostrato che 19 intervalli sono sufficienti per ottenere discreti risultati se la forma d'onda del grafico è distribuita con una certa regolarità nell'ambito del periodo; è chiaro però che maggiore è il numero degli intervalli e più elevata può essere la precisione nella determinazione dello spettro di frequenza del fenomeno in esame.

Dopo aver diviso un periodo in 19 intervalli andiamo a misurare, per ciascuno dei 20 punti, che valore assume il grafico nella scala proposta e formiamo la funzione di tabella con i **campioni** rilevati:

x	a(x)
1	0
2	.16
3	.27
4	.34
5	.38
6	.40
7	.44
8	.56
9	.68
10	.86
11	.98
12	.98
13	.87
14	.30
15	- .14
16	- .34
17	-. 42
18	- .36
19	- .20
20	- .04

Ricavata la funzione di tabella si devono impostare i 3 coefficienti della serie di Fourier in base alla formula di integrazione esposta al paragrafo 7.10 che qui riportiamo:

$$I = (.5\, a(1) + a(2) + a(3) + ... + a(M - 1) + .5\, a(M)) (f - i) / (M-1)$$

in cui i simboli si devono adattare in base al nostro esempio:

-al posto di I si scriveranno, in tre espressioni distinte, A'_0; B'_n A'_n
-al posto di (f - i) si scriverà l'intervallo dell'onda in esame pari a 2π
-M resta il numero massimo dei campioni della funzione di tabella, nel nostro esempio M = 20
-a(x) resta il singolo elemento della funzione di tabella, da a(1) ad a(20)

in base alle definizioni dei coefficienti date nel paragrafo 8.2 si mostrano, a scopo didattico ed in forma estesa, le espressioni che dovranno essere implementate nel programma di calcolo automatico:

$$A'_0 = (1/ 2 \pi) \; [.5\, a(1) + a(2) + a(3) + ... + a(19) + .5\, a(20)] (2 \pi) / 19$$

$$A'_n = (1/\pi) [.5\, a(1) \, Cos\, (nwt_1) + a(2)\, Cos\, (nwt_2) + ... + .5\, a(20)\, Cos\, (nwt_{20})] (2 \pi / 19)$$

$$B'_n = (1/\pi) [.5\, a(1)\, Sen\, (nwt_1) + a(2)\, Sen\, (nwt_2) + ... + .5\, a(20)\, Sen\, (nwt_{20})] (2 \pi / 19)$$

$$C'_n = (A'^2_n + B'^2_n)^{1/2}$$

in cui il valore di n rappresenta la ennesima frequenza dello spettro dopo la quale si ritiene che le ampiezze delle componenti spettrali siano irrilevanti e pertanto da non calcolare; nel nostro esempio fissiamo n =10.
Il calcolo dei coefficienti non richiede precisioni molto elevate, ai fini pratici sono sufficienti tre decimali; pertanto, onde evitare difficoltà di lettura dei dati presentati si adotta una particolare istruzione che limita il numero delle cifre visualizzate. La prima istruzione di questo tipo utilizzata nel programma è così strutturata:

PRINT "A'_o"; **USING "####.###"** ; H

essa prevede la parte intera a 4 cifre (####), e la parte decimale a 3 cifre (###).
Il programma utilizza inoltre una nuova versione dell'istruzione LOCATE detta a riga variabile, istruzione utile per scrivere i dati di uscita su tre colonne ed utilizzare al meglio lo schermo del P.C. l'istruzione è impiegata tre volte nella forma:

LOCATE (u) , 20 :
LOCATE (u) , 40 :
LOCATE (u) , 60 :

dove u, variabile da 1 a p, posiziona i dati l'uno dopo l'altro su righe successive, rispettivamente per le colonne 20; 40; 60.

Vediamo quindi compilato e commentato il programma che esegue il computo automatico, sia dei coefficienti A'_o; A'_n; B'_n, in base alle formule di integrazione sopra esposte, sia delle ampiezze C'_n delle componenti frequenziali o **righe dello spettro** del fenomeno periodico tracciato in figura 33. Essendo il programma a carattere generale prevede l'introduzione del valore di M (nel nostro esempio M = 20).

```
CLS  ' pulisce lo schermo ad inizio operazioni

DIM a(255) ' dimensiona il massimo indice della matrice

INPUT "n"  ;  n    ' richiesta ingresso numero n  righe da calcolare

INPUT "M"  ; M    ' richiesta ingresso numero M elementi della matrice

FOR  x = 1  TO  M  ' comanda impostazione matrice da x=1 a x=M

PRINT "a"   ;  x    ' visualizza a(x) per indicare l'elemento matrice da inserire

INPUT a( x )  ' visualizza il carattere ? accanto al quale digitare il valore di a(x) specificato nell'istruzione precedente

NEXT x   ' rimanda all'istruzione FOR x= per completare la matrice

CLS  ' pulisce lo schermo dai dati digitati

FOR  u = 1  TO  n   ' comanda il calcolo per coppie da A'1 ; B'1 a  A'n ; B'n

FOR  x = 1  TO  M  ' comanda il richiamo degli elementi di matrice da a(1) ad a(M)

nwt = ( 6.283185  /  ( M - 1 ) ) * u * ( x - 1 )   ' calcolo dell'argomento per le funzioni trigonometriche utilizzate nelle
         ' due  istruzioni seguenti (si osservi che la x è diminuita di 1 perché il calcolo delle funzioni       &
```

```
                    ' trigonometriche deve iniziare con argomento a valore zero)

IF x = 1 THEN k = .5 ELSE IF x = M THEN k = .5 ELSE k = 1
                    ' impone k= .5 per il primo e l'ultimo prodotto in base alla regola dei trapezi

B = k * a( x ) * SIN( nwt ) * ( 2 / ( M - 1 ) )  ' prodotto secondo il seno per il calcolo dei
                    ' termini da mettere in sommatoria per la computazione di B'n

C = k * a( x ) * COS( nwt ) * ( 2 / ( M - 1 ) )  ' prodotto secondo il coseno per il calcolo dei
                    ' termini da mettere in sommatoria per la computazione di A'n

T = T + C   ' sommatoria per il calcolo di A'n

Q = Q + B   ' sommatoria per il calcolo di B'n

D = k * a( x ) * ( 1 / ( M - 1 ) ) ' prodotto per calcolo termini da mettere in sommatoria per computazione di A'o

H = ( D + H )  ' sommatoria per il calcolo di A'o

NEXT x   ' rimanda all'istruzione FOR x=... per il richiamo dei successivi elementi di matrice

Cn = SQR ( T ^ 2 + Q ^ 2 )  ' calcolo dell'ampiezza della riga Cn

IF u = 1 THEN PRINT "A'o" ; USING "####.###" ; H
                    ' presenta il dato calcolato di A'o a 4 interi e 3 decimali

LOCATE ( u ),20 : PRINT "A'" ; u ; USING "####.###"; T
                    ' presenta il dato calcolato di A'n a 4 interi e 3 decimali su riga (u) colonna 20

LOCATE ( u ),40 : PRINT "B'" ; u ; USING "####.###"; Q
                    ' presenta il dato calcolato di B'n a 4 interi e 3 decimali su riga (u) colonna 40

LOCATE ( u ),60 : PRINT "C'" ; u ; USING "####.###"; Cn
                    ' presenta il dato calcolato di C'n a 4 interi e 3 decimali su riga (u) colonna 60

Q = 0     ' si azzerano le memorie Q ; T ; H ; Cn prima di ripetere i calcoli per il successivo valore di u
T = 0
H = 0
Cn = 0

NEXT u  ' rimanda all'istruzione FOR u= ... per il calcolo di una nuova coppia di A'n ; B'n.
```

se premiamo F5 il programma chiede:

```
                    il numero delle righe che si vogliono calcolare n ? 10
                    il numero degli elementi che formano la matrice M ? 20
                    i valori di
                      a1
                      ? 0
                      a2
                      ?.16
                      a3
                      ?.27
                      a4
                      ?.34
     valori inseriti  .......  secondo la funzione di tabella
                      a20
                      ?-.04
```

introdotto l'ultimo dato a20 = - .04 il programma presenta i dati calcolati

Ao 0.302	A' 1 -0.497	B' 1 0.223	C' 1 0.545
	A' 2 0.179	B' 2 0.240	C' 2 0.299
	A' 3 0.019	B' 3 -0.068	C' 3 0.070
	A' 4 -0.033	B' 4 0.010	C' 4 0.035
	A' 5 0.019	B' 5 0.018	C' 5 0.026
	A' 6 0.006	B' 6 -0.016	C' 6 0.017
	A' 7 -0.020	B' 7 0.004	C' 7 0.021
	A' 8 0.012	B' 8 0.005	C' 8 0.013
	A' 9 -0.006	B' 9 -0.006	C' 9 0.008
	A' 10 -0.006	B' 10 0.006	C' 10 0.008

I risultati presentati mostrano le 10 coppie dei coefficienti A'_n; B'_n, l'ampiezza delle righe C'_n ad essi corrispondenti ed il valore di A_o.
Si osserva che le righe dello spettro di frequenza decrescono rapidamente e che la 6ª armonica ha una ampiezza di circa 1/30 rispetto alla riga a frequenza base; ciò conferma, in questo caso, che i 10 termini che abbiamo posto a calcolo sono più che sufficienti per rappresentare lo spettro del fenomeno periodico analizzato. Il valor medio dell'onda $A_o = 0.302$ è positivo e sensibilmente diverso da zero, ciò è conforme con il profilo del fenomeno che è sviluppato in prevalenza nel campo dei valori positivi.
Quando un'analisi di questo tipo viene espletata, data la indubbia complessità dell'impostazione e del calcolo, resta qualche ragionevole dubbio sulla possibilità di aver commesso qualche errore tale da inficiare i risultati finali; è possibile un controllo dell'elaborato applicando il programma per la ricostruzione del fenomeno periodico, dati i coefficienti A_o; A_n; B_n, così come mostrato nel paragrafo 8.3, la verifica dei dati che abbiamo ottenuto in questo esercizio sarà oggetto del paragrafo 8.7.

8.6 Aspetto fisico dell'analisi frequenziale

L'analisi frequenziale sviluppata nell'esercizio del paragrafo 8.5 è servita per calcolare le componenti armoniche del fenomeno periodico tracciato in figura 33; i risultati che abbiamo ottenuto, pur espressi in termini di righe dello spettro di frequenza, non contengono alcuna indicazione relativa ai valori delle frequenze che compongono l'onda oggetto dell'analisi. In effetti la serie di Fourier è applicata alla forma dell'onda e non al suo valore di frequenza base f_o che può essere qualsiasi.
E' bene chiarire questo importante aspetto fisico della cosa con due esempi numerici molto semplici: supponiamo che l'onda analizzata abbia un periodo della durata di 1 millesimo di secondo (T = 0.001 Sec.), in questo caso il valore di f_o è $f_o = 1 / T = 1 / 0.001 = 1000$ Hz .
Dato che il calcolo ha mostrato che sono presenti tutti e dieci i valori di C'_n (infatti nessun valore di C_n è uguale a zero) ciò significa che anche tutte le componenti armoniche pari e dispari sono presenti; se la frequenza base **(frequenza fondamentale)** è di 1000 Hz le 10 componenti frequenziali calcolate avranno i seguenti valori:

riga	ampiezza	frequenza Hz
C'1	0.545	1000
C'2	0.299	2000
C'3	0.070	3000
C'4	0.035	4000

C'5	0.026	5000
C'6	0.017	6000
C'7	0.021	7000
C'8	0.013	8000
C'9	0.008	9000
C'10	0.008	10000

Se invece la durata del periodo del fenomeno è ad esempio 20 millesimi di secondo, T = 20 / 1000 Sec. $f_0 = 1 / (20 / 1000) =$ 50 Hz e lo spettro sarà:

riga	ampiezza	frequenza Hz
C'1	0.545	50
C'2	0.299	100
C'3	0.070	150
C'4	0.035	200
C'5	0.026	250
C'6	0.017	300
C'7	0.021	350
C'8	0.013	400
C'9	0.008	450
C'10	0.008	500

Perciò, ferma restante la forma del fenomeno periodico, l'analisi armonica condotta su di esso può esprimere indifferentemente qualsiasi spettro di frequenza in dipendenza della durata del periodo.
Una parola deve essere spesa anche per quanto riguarda le ampiezze delle righe dello spettro; i calcoli di C_n conducono ad una serie di valori che devono essere interpretati correttamente; se ci riferiamo sempre alla tabella sopra riportata dobbiamo tenere presente che i valori di ampiezza che vi compaiono sono da riferirsi alla riga a valore più elevato, nel nostro caso al valore 0.545, perciò le ampiezze di tutte le altre componenti possono essere rapportate alla prima; la tabella può cioè essere normalizzata dividendo tutti i valori per 0.545 in modo da mettere in evidenza i rapporti tra le diverse ampiezze. Così facendo si ottiene la nuova tabella:

riga	ampiezza normalizzata	frequenza Hz
C'1	1	50
C'2	0.548	100
C'3	0.128	150
C'4	0.064	200
C'5	0.047	250
C'6	0.031	300
C'7	0.038	350
C'8	0.023	400
C'9	0.014	450
C'10	0.014	500

Nell'esempio che abbiamo fatto la riga di ampiezza maggiore è risultata essere quella relativa alla frequenza fondamentale, ciò non è una regola, la riga di ampiezza maggiore può risultare una qualsiasi dello spettro in dipendenza della forma dell'onda.

Con i valori di C_n normalizzati si possono tracciare gli spettri a righe impiegando il programma che mostriamo e commentiamo:

```
SCREEN 9                      ' Il programma inizia con la stesura di 11 istruzioni per
FOR x = 0 TO 460 STEP 23      ' la formazione di un sistema di assi cartesiani con
FOR y = 0 TO 160 STEP 2       ' reticolo rettangolare adattato alla presentazione
PSET (x, y), 7                ' dei valori normalizzati di Cn (con valore massimo
NEXT y                        ' di Cn = 1); il reticolo è del tipo rettangolare ad un
NEXT x                        ' solo quadrante, è diviso in 20 intervalli sull'asse x
FOR y = 0 TO 160 STEP 16      ' in modo da consentire la presentazione di 20 righe
FOR x = 0 TO 460 STEP 3       ' è diviso in 10 intervalli sull'asse y in modo da
PSET (x, y), 7                ' facilitare i raffronti tra le ampiezze delle righe
NEXT x
NEXT y
LINE ( 0, 0 )-( 0, 160 )        ' ASSE  Y  ad   1 quadrante
LINE ( 0, 160 )-( 460, 160 )  ' ASSE  X  ad   1 quadrante
DIM a(20)  ' dimensiona matrice volatile per accogliere al massimo 20 valori di Cn
LOCATE 1, 66: INPUT "n=" ; n  ' richiesta del numero n delle righe da presentare
FOR x = 1 TO n  ' impostazione richiesta automatica matrice valori Cn
LOCATE 2, 66: PRINT "C"; x  ' presentazione simbolo Cn per l'istruzione seguente
LOCATE 3, 66: INPUT a( x )  ' presentazione del simbolo  ?  per richiesta valore Cn
NEXT x   ' rimanda all'istruzione FOR x= ... per l'introduzione dei successivi valori di Cn
FOR x = 1 TO n  ' comando per l'esplorazione della matrice volatile
Cn = a( x )  ' estrazione elemento di matrice scandito dall'istruzione precedente
LINE ( 23 * x , 160 ) - ( 23 * x , 160 - 160 * Cn ), 14  ' traccia le n righe a distanza di un intervallo asse x
NEXT x  ' rimanda alla seconda istruzione FOR x= per proseguire l'esplorazione della matrice
```

Il programma si prova con i valori normalizzati calcolati nell'ultima tabella; premendo F5 si ha la comparsa di un piccolo reticolo rettangolare sulla sinistra in alto dello schermo e compare sulla destra la richiesta del numero delle righe e la richiesta dei valori di Cn:

```
n= ? 10
  C 1
  ? 1
   .......
```

e di seguito cancellando
con le frecce si inseriscono
gli altri 9 valori di Cn

dopo l'introduzione dell'ultimo valore si ha la presentazione del reticolo con le 10 righe dello spettro in colore giallo come mostra la figura 35.

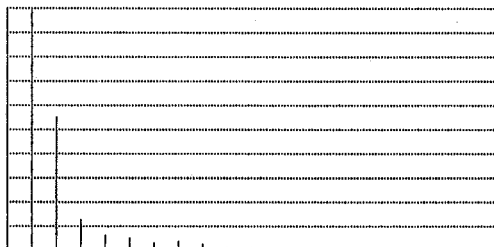



Figura 35
Righe dello spettro del fenomeno periodico di figura 33

8.7 Metodo per il controllo dell'analisi frequenziale

Il metodo per il controllo dell'analisi frequenziale, condotta mediante l'applicazione della serie di Fourier ad un fenomeno periodico, è già stato sperimentato nel paragrafo 8.3 per la verifica di dati computati analiticamente; in questo caso applicheremo il programma di calcolo per il controllo dei dati elaborati nell'esercizio precedente, il lavoro servirà come base generale per la verifica dei coefficienti di una serie di Fourier derivata da un'onda rappresentata da un tracciato.
L'applicazione del programma che ora riproponiamo in forma lievemente modificata, conduce ad un tracciato che, se il metodo non fosse approssimato, dovrebbe essere esattamente sovrapponibile alla curva di figura 33, dato però che tutte le nostre procedure sono necessariamente approssimate il tracciato sarà tanto più vicino all'onda analizzata quanti più campioni sono stati presi in fase di formazione della funzione di tabella.

L'elenco dei dati da verificare, elaborato nel paragrafo 8.5, è costituito dal termine Ao e da 10 coefficienti; da A'1 ad A'10, da B'1 a B'10, come da tabulato sotto riportato:

Ao 0.302	A'1 -0.497	B'1 0.223
	A'2 0.179	B'2 0.240
	A'3 0.019	B'3 -0.068
	A'4 -0.033	B'4 0.010
	A'5 0.019	B'5 0.018
	A'6 0.006	B'6 -0.016
	A'7 -0.020	B'7 0.004
	A'8 0.012	B'8 0.005
	A'9 -0.006	B'9 -0.006
	A'10 -0.006	B'10 0.006

questi dati devono essere inseriti nel programma dopo aver scelto, dato il tipo di grafico da presentare, un sistema di assi cartesiani a soli due quadranti. Inoltre, dato che l'asse delle ascisse dell'onda originale è stato diviso in 19 intervalli, è opportuno, per una più facile comparazione delle curve, l'onda originale e l'onda ricostruita, che il fenomeno venga presentato su 19 intervalli del reticolo; per fare ciò si deve calcolare il valore di K1 dell'istruzione PSET come segue:

$$K1 = (460 / 20) \cdot 19 / 6.28 = 69.58$$

```
LINE ( 0 , 160 ) - ( 460 , 160 ) ' comanda il tracciamento dell'asse delle ascisse (asse X)
                                 ' per coordinate a 2 quadranti (colore = bianco luminoso)

LINE ( 0 , 0 ) - ( 0 , 320 ) ' comanda il tracciamento dell'asse delle ordinate ( asse Y)
                             ' per coordinate a 2 quadranti (colore = bianco luminoso)

DIM A(255) , B(255) ' impostazione delle dimensioni di due matrici volatili per accogliere
                    ' i coefficienti A'n ; B'n

LOCATE 10, 66: INPUT "n" ; k ' richiesta del numero n degli addendi con i quali si vuole realizzare
                             ' la sommatoria della serie di Fourier , il valore coincide con il numero
                             ' delle coppie di coefficienti disponibili, tutte le istruzioni LOCATE sono
                             ' dimensionate per far comparire le richieste all'esterno del reticolo

LOCATE 11, 66: INPUT "A'o" ; Ao ' richiesta di introduzione del valore di A'o

FOR n = 1 TO k ' comando automatico per richiesta n valori di matrice per le due matrici

LOCATE 14, 66: PRINT "A'" ; n ' stampa il simbolo A'n per la successiva istruzione

LOCATE 15, 66: INPUT A( n ) ' stampa il simbolo ? per l'introduzione del valore di A'(n)
                            ' relativo alla prima matrice

LOCATE 18, 66: PRINT "B'" ; n ' stampa il simbolo B'n per la successiva istruzione

LOCATE 19, 66: INPUT B( n ) ' stampa il simbolo ? per l'introduzione del valore di B'(n)
                            ' relativo alla seconda matrice

NEXT n ' rimanda all'istruzione FOR n = ... per completare le due matrici

FOR wt = 0 TO 6.28 STEP .00628 ' fissa la variabilità dell'argomento dei termini trigonometrici
                               ' per le componenti Coseno e Seno della sommatoria della serie
                               ' con incremento di wt tale da ottenere 1000 punti di calcolo

C = Ao ' pone il valore di C uguale ad A'o per sommarlo alla serie

FOR n = 1 TO k ' comando automatico per la lettura dei valori delle due matrici A'(n) ; B'(n)

C = C + A( n ) * COS( n * wt ) + B( n ) * SIN( n * wt ) ' sommatoria progressiva in base ai prodotti
                                                        ' A'n Cos(n wt) e B'n Sen(n wt) secondo lo sviluppo
                                                        'della serie di Fourier

NEXT n ' rimanda all'istruzione FOR n= ... per la scansione delle due matrici

PSET ( 69.58 * wt , 160 - 160 * C ), 14 ' comanda il tracciamento del grafico della funzione ricostruita

NEXT wt ' rimanda all'istruzione FOR wt= per il calcolo di un successivo punto della funzione ricostruita
```

Compilato il programma non resta che inserire:

-il numero delle coppie dei coefficienti (n = 10)
-il valore di A_0
-i valori delle coppie An; Bn elencati nella tabella:

F5

```
n? 10
A0 ? .302
A1
? - 0.497
B1
? 0.223
.................
```

si procede, dopo cancellazione dei dati precedenti con i tasti freccia destra freccia sinistra, ad inserire tutte le restanti 9 coppie di coefficienti .

Dopo la decima coppia si ha la presentazione grafica della ricostruzione del periodo dell'onda come mostrato in figura 36.

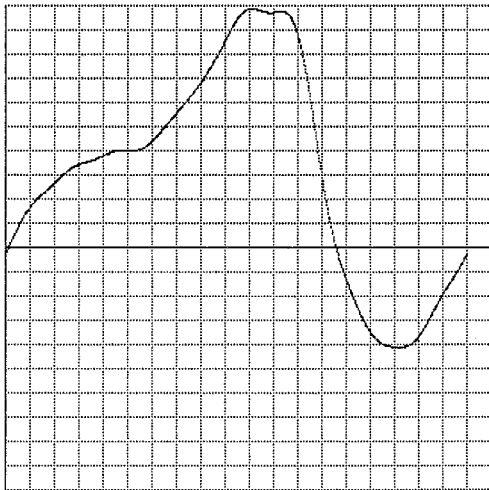



Figura 36
Andamento ricostruito del fenomeno periodico di figura 33

Riportando sul grafico di figura 36 i punti del grafico dell'onda analizzata si osserva che la ricostruzione operata è molto accurata, ciò assicura che i valori dei coefficienti della serie di Fourier che sono stati calcolati in fase di analisi frequenziale sono corretti. Il confronto tra i due grafici è mostrato in figura 37.
Il risultato ottenuto mostra la potenzialità del programma per il controllo dell'analisi frequenziale.
E' utile che il lettore si eserciti con questa procedura di calcolo impiegando nuovi tabulati in Ao, A1, B1 ricavati, applicando il metodo della serie approssimata di Fourier sviluppato nel paragrafo 8.5, da forme d'onda disegnate a tavolino.

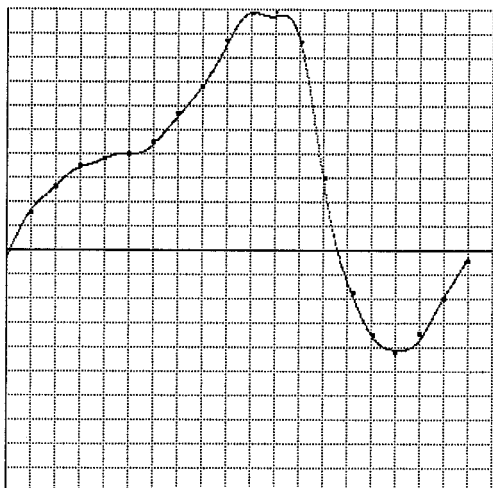



Figura 37
Confronto tra onda originale
e onda ricostruita

8.8 L'integrale o trasformata di Fourier

Per la determinazione dello spettro frequenziale di fenomeni non periodici che si presentano sotto forma di singoli impulsi molto distanziati temporalmente tra loro è disponibile un algoritmo detto **trasformata o integrale di Fourier**. Gli impulsi analizzabili con questo algoritmo possono essere caratterizzati, sia dalla presenza al loro interno di porzioni di onde periodiche, sia da nessuna modificazione interna. La differenza fondamentale tra la serie di Fourier e l'integrale ora menzionato risiede nel fatto che la prima esprime l'onda periodica f(wt) mediante la sommatoria di un numero infinito di componenti frequenziali, ciascuna individuabile da un numero intero n, mentre il secondo individua lo spettro di un fenomeno impulsivo mediante una funzione continua dipendente dalla frequenza.
La serie di Fourier si estende, nel campo delle frequenze, in componenti che possono comprendere come valore minimo la frequenza fondamentale dell'onda per poi crescere soltanto, indefinitamente, nel campo delle frequenze più elevate della fondamentale.
L'integrale di Fourier si estende con continuità, nel campo delle frequenze, per valori sia superiori che inferiori alla frequenza eventualmente contenuta nell'impulso stesso.
In questo paragrafo tratteremo dell'integrale di Fourier al solo scopo di chiarire alcuni elementi necessari alla particolare implementazione di un importante algoritmo in Qbasic, l'argomento sarà perciò soltanto accennato senza alcuna dimostrazione analitica.
La legge matematica che esprime lo spettro di un fenomeno impulsivo è data da:

$$G(w) = \int_{-\infty}^{+\infty} F(t)\ e^{-j\,wt}\ dt$$

in cui
G(w) rappresenta la funzione della frequenza risultato dell'integrale di Fourier
F(t) rappresenta la funzione del tempo che definisce il fenomeno impulsivo da analizzare

$e^{-j\,wt} = \text{Cos } wt - j \text{ Sen } wt$ è l'operatore trigonometrico complesso

L'algoritmo dell'integrale di Fourier non invita certo il lettore, non allenato a questo tipo di sviluppi, a cimentarsi per il calcolo dello spettro di un impulso; anche in questo caso però, come per la serie di Fourier, sono disponibili nutriti elenchi di integrali già sviluppati che, se necessario, possono semplificare il lavoro teorico. A noi non interessa procedere in questo senso, una volta mostrata la forma risolutiva per le ragioni che vedremo, sarà sufficiente riportare un esempio già risolto che ci servirà a titolo di riscontro nel controllo di ciò che andremo ad implementare in Qbasic. Consideriamo pertanto un fenomeno impulsivo di tipo cosinusoidale quale quello mostrato in figura 38

Figura 38
Impulso cosinusoidale

La funzione del tempo che lo definisce è:

$$F(t) = \begin{cases} E \text{ Cos } w_0 t \text{ per } 0 \leq t \leq T \\ 0 \text{ per } t > T \end{cases}$$

Il significato di questa funzione è molto chiaro; dato un fenomeno impulsivo ad andamento cosinusoidale di frequenza f_0, ampiezza E, durata T, che si ipotizza piazzato con il fronte di salita al tempo $t = 0$ si ha:

F(t) assume il valore $E \text{ Cos } w_0 t$ per un tempo t compreso tra 0 e T

F(t) assume il valore 0 per tutto il tempo superiore alla durata T del fenomeno dove $w_0 = 2\,\pi\, f_0$ (pulsazione del fenomeno ondulatorio)

Applicando l'integrale di Fourier a questa funzione del tempo abbiamo la corrispondente funzione della frequenza che esprime l'ampiezza (modulo di G(w)) dello spettro dell'impulso:

$$|G(w)| \cong E\,T/2 \left| \frac{\text{Sen } (w - w_0)\,T/2}{(w - w_0)\,T/2} \right|$$

in cui $w = 2\,\pi\, f$ è la variabile indipendente .
L'espressione rappresenta il valore assoluto di una funzione del tipo Sen x / x che già abbiamo trattato nel paragrafo 2.21. Come si vede il modulo di G(w), spettro del fenomeno impulsivo, è una funzione della variabile indipendente w e come tale rappresentabile da una curva continua.

L'espressione mostra che quando la variabile indipendente w assume il valore della pulsazione w_0 sia il numeratore che il denominatore diventano zero dando luogo ad una forma indeterminata che, così come già visto nel paragrafo sopra citato, calcolata al limite vale 1; ciò evidenzia che il livello massimo dell'ampiezza dello spettro di frequenza dell'impulso coincide con la frequenza f_0 relativa all'onda contenuta nell'impulso.
E' interessante ora tracciare il grafico dello spettro che abbiamo esaminato prendendo ad esempio un impulso con le seguenti caratteristiche:

-durata dell'impulso cosinusoidale T = 0.002 Sec.
-frequenza dell'ondulazione contenuta nell'impulso f_0 = 10000 Hz
-pulsazione angolare dell'ondulazione $w_0 = 2\pi f_0$
-ampiezza dell'impulso E = 1000
-coefficiente E T / 2 = 1

da analizzare, a scopo dimostrativo, in un campo frequenziale che inizia da 2000 Hz sotto la frequenza dell'ondulazione e finisce a 2000 Hz sopra tale frequenza; corrispondente ad un intervallo compreso tra 8000 Hz e 12000 Hz.
Con i valori assunti possiamo iniziare scrivendo | G(w) | in Qbasic:

| G(w)| = y =

= ABS ((SIN ((6.28 * f - 6.28 * 10000) * 0.002/2) / ((6.28 * f - 6.28 * 10000) * 0.002/2)

e quindi compilare il programma per il tracciamento del modulo di G(w):

-si calcola il valore k1 per l'istruzione PSET k1 = 460 / (12000 - 8000) = .115
-si compone la variabile x di PSET nella forma (f - 8000) * .115
-si fissa la presentazione su assi cartesiani ad 1 quadrante
-l'asse delle ascisse (asse della variabile indipendente f) è suddiviso in 20 intervalli da 200 Hz ciascuno
-l'asse delle ordinate (asse della variabile dipendente |G(w)|) è suddiviso in 20 intervalli da .05

```
LINE ( 0 , 320 ) - ( 460 , 320 )  ' assi cartesiani ad 1 quadrante indicati per una funzione che si sviluppa
                                  ' soltanto nel campo delle ascisse e delle ordinate positive

LINE ( 0 , 0 ) - ( 0 , 320 )      ' (colore = bianco luminoso)

LOCATE 23 , 66 : PRINT "Y- Div. = .05"  ' produce la scritta    Y-Div. = .05  nell'angolo
                                        ' basso a destra dello schermo

LOCATE 22 , 66 : PRINT "X- Div. = 200 Hz"  ' produce la scritta X-Div .=200Hz nell'angolo
                                           ' basso a destra dello schermo

FOR f= 8000 TO 12000 STEP 1.1
        ' campo di variabilità della frequenza di w= (2 π f) con incremento
        ' di 1.1 Hz ( incremento non intero per evitare la forma indeterminata)
        ' ( pari circa a 3600 punti di calcolo)

y = ABS ( ( SIN ( ( 6.28 * f - 6.28 * 10000 ) * 0.002/2 ) ) / ( ( 6.28 * f - 6.28 * 10000 ) * 0.002/2 ) )
                                                        ' funzione | G(w)| da computare

PSET ( ( f - 8000 ) *.115 , 320 - 320 * y ),14   ' comanda il tracciamento del grafico in giallo

NEXT f  ' comanda il programma al ritorno automatico all'istruzione FOR f= .. ecc.
```

sullo schermo si ha la presentazione del reticolo e della curva caratteristica dello spettro dell'impulso cosinusoidale così come riportato in figura 39 dalla quale si osserva:

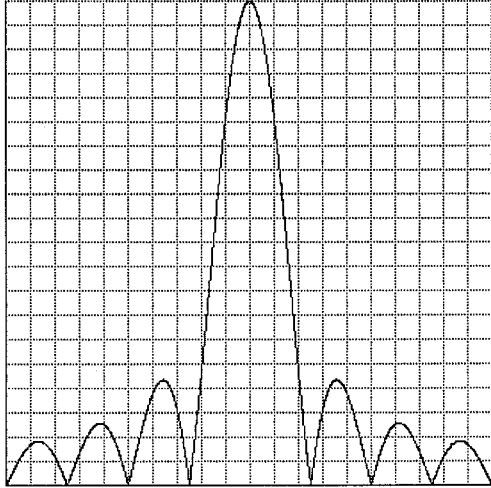



Figura 39
Spettro dell'impulso cosinusoidale

-La curva mostra la continuità dello spettro del fenomeno impulsivo, contrariamente alle discontinuità degli spettri a righe frutto dell'analisi armonica dei fenomeni periodici.
-La curva è tracciata tra l'origine degli assi, coincidente con la frequenza 8000 Hz, e la fine dell'asse delle ascisse coincidente con la frequenza 12000 Hz.
-La curva ha il suo valore massimo al centro dell'asse delle ascisse in coincidenza della frequenza di 10000 Hz, tale frequenza caratterizza il fenomeno periodico contenuto nell'impulso.
-I primi zeri della $|G(w)|$ sono collocati rispettivamente alle frequenze di 9500 Hz e 10500 Hz, ad una distanza frequenziale pari a 10500 - 9500 = 1000 Hz . Questo intervallo è calcolabile in base alla durata dell'impulso secondo l'espressione $2 / T = 2 / .002 = 1000$ Hz.
-Lo spettro si estende senza soluzione di continuità anche per frequenze esterne al tracciato.

8.8.1 La trasformata di Fourier e la collocazione degli spettri

Per comprendere meglio come possono essere collocati, nel campo delle frequenze, gli spettri dei fenomeni impulsivi è di aiuto lo sviluppo dell'integrale di Fourier relativo ad un semplice impulso a caratteristica rettangolare come quello mostrato in figura 40.
Questo impulso è definito da una legge matematica simile a quella descritta in precedenza per l'impulso a contenuto oscillante:

$$F(t) = \begin{cases} E & \text{per } 0 \leq t \leq T \\ 0 & \text{per } t > T \end{cases}$$

Il significato di questa funzione è semplice; dato un fenomeno impulsivo semplice di ampiezza E e durata T, che si ipotizza piazzato con il fronte di salita al tempo t = 0 si ha:

F(t) assume il valore E per un tempo t compreso tra 0 e T
F(t) assume il valore 0 per tutto il tempo superiore alla durata T del fenomeno

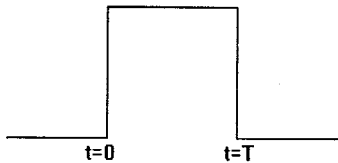



Figura 40
Impulso rettangolare

Applicando l'integrale di Fourier a questa funzione del tempo abbiamo la corrispondente funzione della frequenza che esprime il modulo di G(w) dello spettro dell'impulso; il risultato dell'integrale ci è fornito dagli appositi elenchi ed è:

$$|G(w)| = E\,T \left| \frac{\text{Sen}(wT/2)}{(wT/2)} \right|$$

L'espressione dello spettro è, come per l'esercizio precedente, del tipo Sen x/x ma senza il valore di w_0 dato che in questo caso all'interno dell'impulso non sono presenti fenomeni oscillatori; in essa compare soltanto la variabile indipendente $w = 2\pi f$.
Questa sostanziale differenza porta la funzione ad avere il suo valore massimo in corrispondenza della frequenza zero e pertanto una collocazione dello spettro dell'impulso completamente diversa dal primo esempio.
Per tracciare il grafico dello spettro dell'impulso rettangolare consideriamo ad esempio un impulso con le seguenti caratteristiche:

-durata dell'impulso T = 0.01 Sec.
- ampiezza dell'impulso E = 100
- coefficiente E T = 1

da analizzare in un campo frequenziale che deve iniziare necessariamente da 0 Hz e finire ad esempio a 1000 Hz.
Con i valori assunti possiamo iniziare scrivendo | G(w) | in Qbasic:

```
| G( w )| = y = ABS ( SIN ( 6.28 * f * .01 / 2 ) / ( 6.28 * f * .01 / 2 ) )
```

e quindi compilare il programma per il tracciamento del modulo di G(w):

-si calcola il valore k1 per l'istruzione PSET k1 = 460 / 1000 = .46
-si compone la variabile x di PSET nella forma f * .46
-si fissa la presentazione su assi cartesiani ad 1 quadrante
-l'asse delle ascisse (asse della variabile indipendente f) è suddiviso in 20 intervalli da 50 Hz
-l'asse delle ordinate (asse della variabile dipendente |G(w)|) è suddiviso in 20 intervalli da .05

```
LINE ( 0 , 320) - ( 460 , 320 )  ' assi cartesiani ad I quadrante indicati per una funzione che si sviluppa
                                  ' soltanto nel campo delle ascisse e delle ordinate positive

LINE ( 0 , 0 ) - ( 0 , 320 )      ' (colore = bianco luminoso)

LOCATE 23 , 66 : PRINT "Y- Div. =.05"   ' produce la scritta  Y-Div. = .05  nell'angolo
                                         ' basso a destra dello schermo

LOCATE 22 , 66 : PRINT "X- Div. = 50 Hz"  ' produce la scritta X-Div.=50Hz nell'angolo
                                           ' basso a destra dello schermo

FOR  f = 0.001  TO  1000  STEP .1
                        ' campo di variabilità della frequenza di w= (2 π f)  con incremento
                        ' di 1Hz ed inizio per f=0.001  onde evitare la forma indeterminata
                        ' ( pari circa a 10000 punti di calcolo )

y = ABS ( SIN ( 6.28 * f * .01 / 2 ) / ( 6.28 * f * .01 / 2 ) )
                                   ' funzione  | G(w)| da computare

PSET ( f * .46 , 320 - 320 * y ),14   ' comanda il tracciamento del grafico in giallo

NEXT f  ' comanda il programma al ritorno automatico all'istruzione FOR f= .. ecc.

                    F5
```

sullo schermo si ha la presentazione del reticolo e della curva caratteristica dello spettro dell'impulso rettangolare così come riportato in figura 41.

-La curva mostra la continuità dello spettro del fenomeno impulsivo, contrariamente alle discontinuità degli spettri a righe frutto dell'analisi armonica dei fenomeni periodici.
-La curva è tracciata tra l'origine degli assi, coincidente con la frequenza 0 Hz, e la fine dell'asse delle ascisse coincidente con la frequenza 1000 Hz.
-La curva ha il suo valore massimo all'inizio dell'asse delle ascisse in coincidenza della frequenza zero.
-Il primo zero della |G(w)| è collocato alla frequenza di 100 Hz, questo valore è calcolabile in base al reciproco della durata dell'impulso: $1 / T = 1 / .01 = 100$ Hz .
-Lo spettro si estende senza soluzione di continuità anche per i valori di frequenza più elevata non compresi nel tracciato.
-Le variazioni di ampiezza dello spettro, per frequenze superiori alla frequenza del primo zero, decrescono indefinitamente .
Con lo sviluppo di questo programma abbiamo completato due esercizi tipici relativi alla trasformata di Fourier; a complemento del lavoro svolto evidenziamo alcune caratteristiche significative relative ai moduli di G(w):

-le funzioni che esprimono |G(w)| sono del tipo | Sen x/x |, sia per l'impulso cosinusoidale che per l'impulso rettangolare

-per entrambi i moduli restano fissi i rapporti tra le ampiezze dello spettro fuori del massimo e l'ampiezza del massimo stesso

-è possibile cambiare, se necessario, i rapporti di cui al punto precedente mediante modifiche calibrate dei profili degli impulsi seguendo un processo particolare che prende il nome di finestra di Tukey .
Il metodo è indicato da Harris nel lavoro USE OF WINDOWS FOR HARMONIC ANALYSIS.

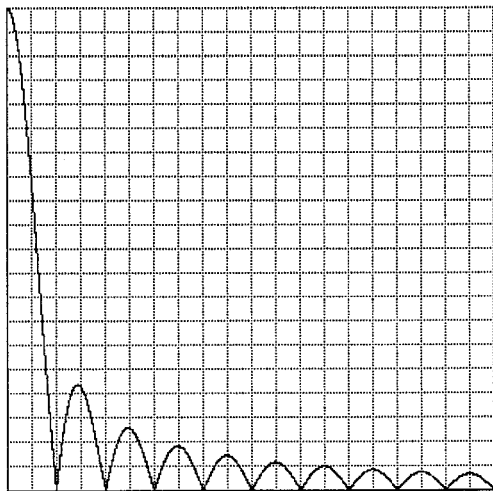



Figura 41
Spettro dell'impulso rettangolare di figura 40

8.9 Metodo di approssimazione per l'integrale di Fourier - la D F T -

Abbiamo constatato, nel paragrafo precedente, che l'applicazione dell'integrale di Fourier richiede la conoscenza analitica della funzione impulsiva della quale si voglia determinare lo spettro di frequenza; quando la funzione del tempo che descrive il fenomeno impulsivo è nota si può tentare il calcolo della trasformata di Fourier, sia con l'ausilio dei già citati elenchi, sia mediante sviluppi analitici molto complicati. Quando poi il fenomeno impulsivo non è definito matematicamente, ma se ne conosce soltanto il profilo, il problema della determinazione dello spettro di frequenza non è affrontabile con i metodi ordinari dell'analisi matematica. In questi casi gioca un ruolo fondamentale un metodo di approssimazione per l'integrale di Fourier, implementabile in Qbasic, che dà modo di risolvere con buoni risultati il problema menzionato.
Il metodo in questione è definito con l'acronimo **D F T** composto dalle iniziali della terminologia inglese **D**iscrete **F**ourier **T**ransform (Trasformata di Fourier per punti discreti).
La DFT consente il calcolo approssimato dello spettro di frequenza di un fenomeno impulsivo quando di esso si conoscano un certo numero di valori d'ampiezza (campioni) rilevati nell'ambito della durata dell'impulso stesso.
Per i nostri scopi l'algoritmo della DFT può essere espresso mediante la sommatoria:

$$G(q f) = \sum_{p=1}^{k} S(p)\ e^{-j 2 \pi q p / N}$$

in cui:
G(qf) rappresenta lo spettro complesso dell'impulso nei q punti calcolati
N rappresenta il rapporto $N = 2F_{max} / DF$ dove F_{max} è la massima frequenza dello

spettro che si vuole analizzare e DF è l'intervallo di frequenza assegnato tra due valori contigui di G(qf).

k è un intero che rappresenta il numero di campioni da rilevare dall'impulso in base alla relazione $k = 2\,T\,F_{max}$ dove T è la durata temporale dell'impulso e F_{max} è la massima frequenza dello spettro che si vuole analizzare

p è il numero di posizione del generico campione variabile da p = 1 a p = k

S(p) rappresenta il generico valore di ampiezza (campione temporale) dei k misurati sull' impulso da analizzare

$e^{-j\,2\pi q p / N}$ è l'operatore trigonometrico complesso

q rappresenta il numero discreto dei punti calcolati dello spettro complesso di frequenza normalmente fissato dal rapporto $q \leq F_{max} / DF$

Il calcolo dei q termini complessi della DFT si svolge come segue:
per p = 1 si esegue il prodotto tra S(1) e l'operatore trigonometrico in cui q = 0; p = 1, si ripete l'operazione per p = 2; q = 0, ..., si ripete infine l'ultimo prodotto per p = k; q = 0, si sommano tra loro i k prodotti e si ottiene il primo valore di G(qf) per q = 0.
Il calcolo viene replicato come il precedente ma per q = 1, in questo caso la sommatoria dei k prodotti fornisce il secondo valore di G(qf) per q = 1.
Si procede in modo analogo per q = 2; q = 3; ... ottenendo rispettivamente il terzo, quarto,.. ecc. valore complesso di G(qf).
La sommatoria della DFT che abbiamo mostrato è in forma complessa e come tale non può essere implementata in Qbasic; per il calcolo dell'ampiezza dello spettro di un fenomeno impulsivo, mediante programma esecutivo della DFT in Qbasic, è necessaria la conoscenza del modulo di G(qf) che esponiamo senza dimostrazione:

$$|G(qf)| = \left[\left(\sum_{p=1}^{k} S(p)\,\mathrm{Cos}\,(2\pi q p / N) \right)^2 + \left(\sum_{p=1}^{k} S(p)\,\mathrm{Sen}\,(2\pi q p / N) \right)^2 \right]^{1/2}$$

in cui

|G(qf)| rappresenta il modulo dello spettro complesso dell'impulso nei q punti calcolati

N rappresenta il rapporto $N = 2F_{max} / DF$ dove F_{max} è la massima frequenza dello spettro che si vuole analizzare e DF è l'intervallo di frequenza assegnato tra due valori contigui di |G(qf)|.

k è un intero che rappresenta il numero di campioni da rilevare dall'impulso in base alla relazione $k = 2\,T\,F_{max}$ dove T è la durata temporale dell'impulso e F_{max} è la massima frequenza dello spettro che si vuole analizzare. Più elevato è il valore di k più preciso è il calcolo dello spettro.

p è il numero di posizione del generico campione variabile da p = 1 ad p = k

S(p) rappresenta il generico valore di ampiezza (campione) dei k misurati sull'impulso da analizzare

q rappresenta il numero discreto dei punti calcolati dello spettro complesso di frequenza normalmente fissato dal rapporto $q \leq F_{max} / DF$

Si deve chiarire subito, ad evitare equivoci, che i q punti calcolati di |G(qf)| **non sono righe** dello spettro ma rappresentano i valori dello spettro continuo che sono stati computati; tra due

qualsiasi di questi valori contigui si possono calcolare, sempre che il numero dei campioni dell'impulso lo permetta, altri valori dello spettro di frequenza che caratterizzano il fenomeno impulsivo.

8.9.1 Implementazione della DFT in Qbasic

Per l'implementazione dell'espressione di |G(qf)| presentiamo un programma appositamente studiato e commentato che sviluppa il calcolo dell'ampiezza dei q valori calcolati dello spettro di un fenomeno impulsivo. Il programma si avvale di una particolare routine di calcolo per la normalizzazione dei valori del modulo di G(qf); con questo metodo si ha la possibilità di tracciare il grafico normalizzato con un massimo di punti pari a q = 255. La presentazione grafica prevede l'impiego di un reticolo quadrato ad un solo quadrante, diviso sulle ascisse e sulle ordinate in 20 intervalli, sul quale non sono tracciati gli assi cartesiani per consentire la visione dei punti che possono cadere sugli estremi.
Al fine di rendere più chiara la stesura del programma, questa è stata suddivisa in 4 sezioni:

-Grafica per tracciamento del solo reticolo
-Procedimento per il calcolo della DFT (con richiesta dati)
-Procedimento per la normalizzazione della |G(qf)| (in tre passi)
-Presentazione grafica della |G(qf)| normalizzata

```
' GRAFICA - per tracciare il solo reticolo-

SCREEN 9

FOR x = 0 TO 460 STEP 23

FOR y = 0 TO 320 STEP 2  ' VERTICALE

PSET (x, y), 7

NEXT y

NEXT x

FOR y = 0 TO 320 STEP 16

FOR x = 0 TO 460 STEP 3  ' ORIZZONTALE

PSET (x, y), 7

NEXT x

NEXT y

LOCATE  9 , 66 : INPUT "Fmax=" ; rr  ' richiesta valore di fondo scala per calibrazione ascisse

ff = rr / 20  ' calcolo Div. ascisse

LOCATE 23 , 64 : PRINT "y-Div.=. 05"

LOCATE 22 , 64 : PRINT "x-Div.= " ; ff ; "Hz"

' PROCEDIMENTO PER IL CALCOLO DELLA DFT con le richieste dati posizionate a destra del 'reticolo

LOCATE 10 , 66 : INPUT "N=" ; N  ' entra il valore N=2Fmax / DF

LOCATE 11 , 66 : INPUT "q=" ; q  ' entra il numero dei punti di |G(qf)| da calcolare  q = Fmax / DF    &
```

```
LOCATE 12 , 66 : INPUT "k=" ;  k   ' entra il numero dei campioni  k= 2 T Fmax   che sono rilevati
                                   ' all'interno dell'impulso

DIM S( k ), u( q ), v( q )  ' dimensionamento matrici volatili  S(k);u(q);v(q)

FOR  p = 1  TO  k   ' istruzione  per inserimento automatico campioni S(p)  da 1 a k

LOCATE 13 , 66 : PRINT "S" ;  p   ' istruzione che visualizza  il simbolo del campione da inserire S(p)

LOCATE 14 , 66 : INPUT S( p )   ' istruzione che visualizza il simbolo ? per l'ingresso di S(p)

NEXT p

FOR  i = 0  TO  q   ' istruzione per il calcolo automatico dei  q  valori di |G(qf)|

FOR  p = 1  TO  k  ' istruzione per l'esecuzione delle due sommatorie in seno e coseno

z = 6.283185   ' valore che ricorre negli argomenti dei termini in seno e coseno

C = S( p ) * COS ( p * i * z / N )   ' calcolo dei termini in coseno

D = S( p ) * SIN ( p *  i * z / N )   ' calcolo dei termini in seno

T = T + C    ' sommatoria di funzione dei termini in coseno

S = S + D    ' sommatoria di funzione dei termini in seno

NEXT p   ' rimanda alla seconda istruzione FOR p=1... per la sommatoria dei termini trigonometrici

u( i ) =  SQR( T ^ 2 + S ^ 2 )  ' calcolo modulo di G(qf)  e formazione matrice u(i)

T = 0   ' pulizia delle memorie
S = 0
v = 0

NEXT i   ' rimanda all'istruzione For=i .. per il calcolo dei  q valori di |G(qf)

'  PROCEDIMENTO  INDIRIZZATO ALLA NORMALIZZAZIONE DI |G(qf)|

' PRIMO PASSO - creazione di una matrice uguale a u(i)

 FOR i = 0 TO q ' formazione della matrice v(i) per la ricerca del massimo di |G(qf)|

v( i ) =  u( i )   ' la matrice v(i) è una copia della matrice u(i)

NEXT i    ' rimanda all'istruzione FOR i = .. per il completamento copia matrice

 ' SECONDO PASSO - isolamento del valore massimo della matrice v(i)=u(i)

FOR  i = 0  TO  q

FOR  r = 0  TO  q

IF  v( i ) <  v( r )  THEN  v( i ) = 0   ' l'isolamento del valore massimo della matrice v(i) ; si ottiene
                                         ' uguagliando a 0 tutti i termini

NEXT r                                   ' della matrice che sono inferiori agli altri

NEXT i                                                                                   &
```

```
' TERZO PASSO -ricerca del massimo della matrice v(i)=u(i)

FOR i = 0 TO q    ' procedimento finale per la
                  ' ricerca del massimo di v(i)

IF v( i ) > 0 THEN v = v( i )  ' max = v

NEXT i

' NORMALIZZAZIONE DELLA IG(qf)I

' la normalizzazione di IG(qf)I si ottiene con il rapporto u(i) / v

FOR i = 0 TO q  ' calcolo del rapporto u(i)/v

h = u( i ) / v  ' IG(qf)I normalizzata

' PRESENTAZIONE GRAFICA della IG(qf)I normalizzata

PSET ( (460 / q) * i , 320 - 320 * h ), 14      ' presentazione grafica dei q punti di IG(qf)I normalizzata

PSET ( (460 / q) * i + 1 , 320 - 320 * h ),14   ' mediante gruppi di 5 punti per meglio evidenziarne

PSET ( (460 / q) * i - 1 , 320 - 320 * h ),14   ' le tracce mediante crocette

PSET ( (460 / q) * i , 320 - 320 * h +1 ),14

PSET ( (460 / q) * i , 320 - 320 * h - 1 ),14

NEXT i  ' rimanda all'ultima istruzione FOR i=.. per il tracciamento dei q punti
```

Per prendere confidenza con il nuovo metodo di calcolo è indispensabile svolgere alcuni esempi numerici sulla base degli elementi in nostro possesso; si suggerisce inizialmente la scelta dei valori di DF e F_{max} secondo le seguenti limitazioni: DF < 1 / (3 T), F_{max} > 5 / T.
Prendiamo in esame un impulso rettangolare le cui caratteristiche siano:

-durata dell'impulso T = 0.01 Sec.
-ampiezza dell'impulso E = 1

Proponiamoci la determinazione dello spettro dell'impulso, tramite DFT, da frequenza 0 a F_{max} = 1000 Hz a passi di frequenza pari a DF = 25 Hz; si calcolano inizialmente i valori di:
k = 2 T F_{max} = 2 · .01 Sec. · 1000 Hz = 20
N = 2 F_{max} / DF = 2 · 1000 / 25 = 80
q = ≤ F_{max} / DF = 1000 / 25 = 40
S(p) da p = 1 a p = k = 20, dato che l'impulso è rettangolare tutti e 20 campioni sono uguali ad E = 1.
Questi dati possono essere inseriti nel programma per il calcolo di IG(qf)I:

```
                F5
Fmax= ? 1000
N= ? 80
q= ? 40
k= ? 20
S 1
? 1
```
...... di seguito fino a S20 cancellando di volta in volta i campioni digitati e già inseriti.

Dopo l'introduzione dell'ultimo campione S(20) si ha la presentazione del reticolo nel quale compaiono, a punti gialli, i 40 valori calcolati per |G(qf)| normalizzata così come mostrato in figura 42, si ha un dato ogni 1000 / 40 = 25 Hz.

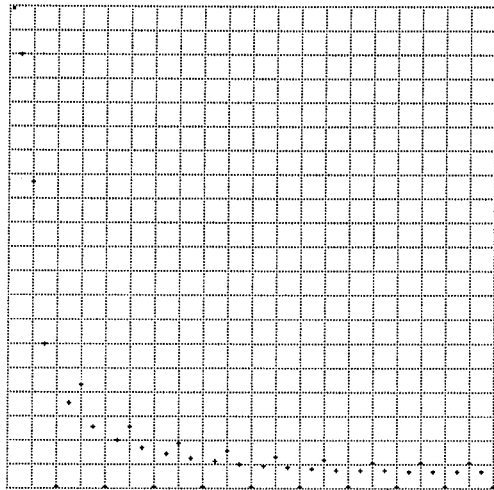



Figura 42
|G(qf)| normalizzata

Per avere un'idea della bontà dell'algoritmo DFT è utile confrontare i risultati ottenuti con questa procedura con dati elaborati analiticamente mediante la trasformata di Fourier; il confronto è reso possibile ricavando dalla figura 42 un tabulato di 20 valori della |G(qf)| normalizzata e riportando, a fianco di questi, i valori determinati analiticamente, per lo stesso tipo di impulso, nell'esercizio del paragrafo 8.8.
Scegliendo i primi 20 valori consecutivi di q ciascuno di essi sarà separato dagli altri di 1000/40 = 25 Hz e la tabella sarà così configurata:

q	f (Hz)	calcolo con DFT |G(qf)|	calcolo analitico |G(qf)|
0	0	1	1
1	25	.9	.9
2	50	.64	.65
3	75	.30	.30
4	100	0	0
5	125	.18	.17
6	150	.21	.21
7	175	.13	.12
8	200	0	0
9	225	.1	.1
10	250	.13	.12

11	275	.08	.07
12	300	0	0
13	325	.07	.07
14	350	.09	.09
15	375	.06	.06
16	400	0	0
17	425	.06	.05
18	450	.08	.07
19	475	.05	.05

Dalla tabella si osserva una ottima corrispondenza tra la |G(qf)| computata con DFT e la |G(qf)| computata per via puramente analitica, questo riscontro evidenzia la bontà della DFT che pertanto può essere applicata con certezza di validi risultati per lo studio dei fenomeni impulsivi non definiti matematicamente.

8.9.2 Esempio applicativo della DFT

Un secondo esempio ci aiuterà ad usare il programma per la DFT nella generalità dei casi; supponiamo di dover eseguire l'analisi di un fenomeno impulsivo quale quello tracciato in figura 43, fenomeno della durata temporale T = .0125 Sec.
Si voglia determinare lo spettro dell'impulso, tramite DFT, da frequenza 0 a Fmax = 1000 Hz a passi di frequenza pari a DF = 25 Hz.
Si devono inizialmente calcolare i valori di:
$k = 2\,T\,F_{max} = 2 \cdot .0125$ Sec. $\cdot$ 1000 Hz = 25
$N = 2\,F_{max} / DF = 2 \cdot 1000 / 25 = 80$
$q = \leq F_{max} / DF = 1000 / 25 = 40$
S(p) da p = 1 a p = k = 25 secondo la tabella seguente, ricavata dalla figura 43, dopo aver suddiviso l'asse delle ascisse in 24 intervalli uguali e misurate le 25 ascisse, S(p), sul disegno:

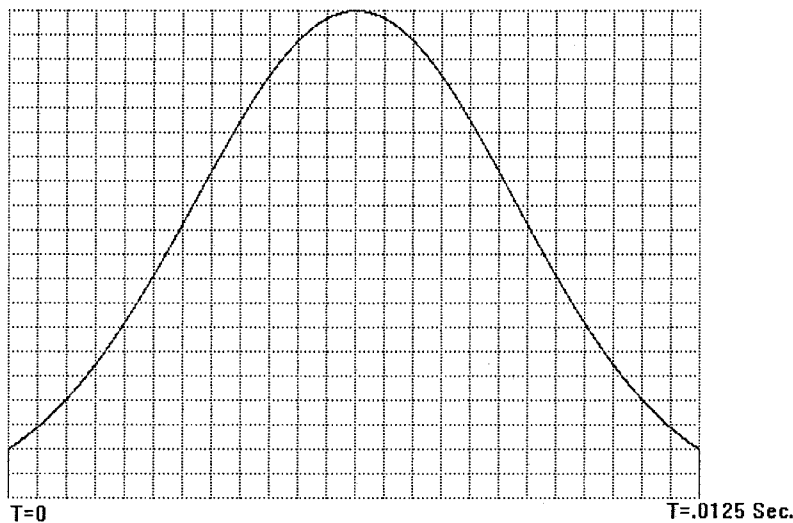



Figura 43
Impulso generico da analizzare

p	S(p)
1	.10
2	.14
3	.20
4	.27
5	.36
6	.46
7	.56
8	.67
9	.77
10	.86
11	.93
12	.98
13	1.00
14	.98
15	.93
16	.86
17	.77
18	.67
19	.56
20	.46
21	.36
22	.27
23	.20
24	.14
25	.10

questi dati possono essere inseriti nel programma per il calcolo di |G(qf)|:

```
            F5

Fmax= ? 1000
N= ? 80
q= ? 40
k= ? 25
S 1
? .10
...... di seguito fino a S25 cancellando
...... di volta in volta i campioni digitati
...... e già inseriti.
......
```

Dopo l'introduzione dell'ultimo campione S(25) si ha la presentazione del reticolo nel quale compaiono, a punti gialli, i 40 valori calcolati per |G(qf)| normalizzata così come mostrato in figura 44; si ha un dato ogni 1000 / 40 = 25 Hz.
Lo spettro dell'impulso è visualizzato dalla frequenza 0 alla frequenza massima di 1000 Hz.
Anche in questo caso, come nell'esempio mostrato nel paragrafo 8.8.1, l'ampiezza massima del modulo di G(w) si ha alla frequenza zero; ciò è dovuto alle caratteristiche dell'impulso analizzato che non contiene fenomeni ondulatori al suo interno.

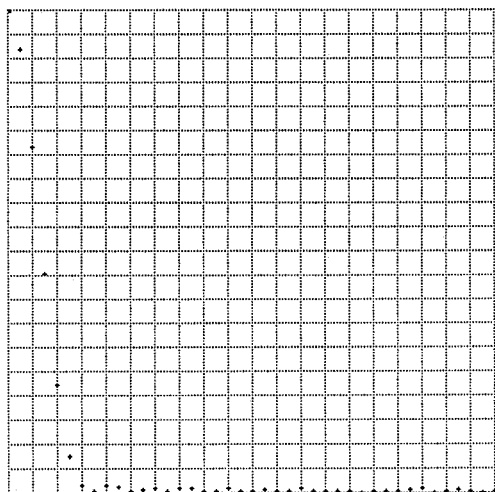



Figura 44
Spettro del fenomeno impulsivo di figura 43 calcolato con DFT

Lo spettro del fenomeno impulsivo che abbiamo ottenuto, diversamente dal precedente esempio, non è verificabile analiticamente dato che non si conosce la legge matematica che governa l'impulso di figura 43; una volta però che il programma è stato testato si può essere ragionevolmente fiduciosi che il risultato ottenuto è significativo.

CAPITOLO 9
GLI ALGORITMI DI CORRELAZIONE

Vanno sotto il nome di algoritmi di correlazione una serie di strumenti matematici che eseguono la ricerca dei legami di interdipendenza tra fenomeni che interessano tutte le scienze.
La possibilità di implementare in Qbasic questi algoritmi offre al lettore un efficace mezzo di indagine in molti campi degli studi e della tecnica.

9.1 La correlazione tra grandezze in numero discreto

Correlazione è la dipendenza reciproca tra due serie di grandezze, l'entità della dipendenza è definita come **coefficiente di correlazione** (C). Se le due serie di grandezze sono strettamente dipendenti l'una dall'altra si ha un elevato coefficiente di correlazione positivo o negativo, se le due serie di grandezze sono poco dipendenti tra loro si ha un modesto valore del coefficiente di correlazione, se infine le due serie di grandezze sono totalmente indipendenti si ha un coefficiente di correlazione nullo.

Il massimo valore del coefficiente di correlazione è dato da $C = 1$ o da $C = -1$
Il minimo valore del coefficiente di correlazione è dato da $C = 0$

Chiariamo la cosa prendendo due serie di grandezze individuate rispettivamente dalle lettere:

A1; A2; ....; An e B1; B2;.....; Bn

calcoliamo le **medie** delle due serie, Am e Bm:

$Am = (A1 + A2 +...+ An) / n$ $\qquad$ $Bm = (B1 + B2 +...+ Bn) / n$

impostiamo il rapporto:

$$C = \frac{(A1-Am) (B1-Bm) + (A2-Am) (B2-Bm) + ...+ (An-Am) (Bn-Bm)}{\left[[(A1-Am)^2 +(A2-Am)^2 + .. +(An-Am)^2] \; [(B1-Bm)^2 +(B2-Bm)^2 +..+(Bn-Bm)^2] \right]^{1/2}}$$

il valore di C esprime il coefficiente di correlazione tra le due serie di valori A1...An; B1;...Bn.
Nei processi statistici sono significativi inoltre i valori detti di **deviazione standard** che per ciascuna delle due serie sono calcolabili con le seguenti espressioni:

$$DsA = \left[\left((A1-Am)^2 + (A2-Am)^2 + ... + (An-Am)^2 \right) / n \right]^{1/2}$$

$$DsB = \left[\left((B1-Bm)^2 + (B2-Bm)^2 + ... + (Bn-Bm)^2 \right) / n \right]^{1/2}$$

Per rendere più tangibile il significato del coefficiente di correlazione è necessario implementare in un programma in Qbasic gli algoritmi che abbiamo ora mostrato ed operare con questi su

particolari serie di valori che possano illustrare come le caratteristiche del loro legame siano esprimibili mediante il valore di C.
Il programma consente il calcolo, sia del valore del coefficiente di correlazione C di cui ci occuperemo, sia dei valori medi Am; Bm e delle deviazioni standard DsA; DsB che possono essere utili a chi si interessa di problemi statistici.
La compilazione del programma richiede una scomposizione delle formule date in modo che risulti più facile trovare le corrispondenze simboliche in linguaggio Qbasic; il commento al programma sotto elencato mette in evidenza questa scomposizione che si avvale delle seguenti memorie di appoggio:

A = sommatoria progressiva dei termini An per il calcolo delle medie Am
B = sommatoria progressiva dei termini Bn per il calcolo delle medie Bm
Ao = differenze alle medie per i termini An
Bo = differenze alle medie per i termini Bn
SP = sommatoria progressiva dei prodotti delle sommatorie
Aq = sommatoria progressiva dei quadrati delle differenze alla media per la serie An
Bq = sommatoria progressiva dei quadrati delle differenze alla media per la serie Bn
RAq = calcolo radice quadrata di Aq
RBq = calcolo radice quadrata di Bq

```
CLS ' pulisce lo schermo

INPUT "n=" ; n ' entra il numero dei termini delle due serie (massimo valore di n =250)

DIM A( n ), B( n ) ' dimensionamento matrici volatili per contenere i valori delle due serie di dati

FOR P = 1 TO n ' comanda richiesta caricamento matrice A(n)

LOCATE 1, 10: PRINT "A" ; P ' richiesta valori della serie An

LOCATE 2, 10: INPUT A( P ) ' caricamento matrice volatile A(n)

NEXT P ' rimanda all'istruzione FOR P... per il successivo valore della serie

FOR P = 1 TO n ' comanda richiesta caricamento matrice B(n)

LOCATE 3, 10: PRINT "B" ; P ' richiesta valori della serie Bn

LOCATE 4, 10: INPUT B( P ) ' caricamento matrice volatile B(n)

NEXT P ' rimanda alla seconda istruzione FOR P... per il successivo valore della serie

FOR i = 1 TO n ' governa il calcolo delle sommatorie progressive per il

A = A + A( i )        ' successivo calcolo    delle medie di An e Bn

B = B + B( i )

NEXT i ' rimanda a FOR i... per il completamento delle sommatorie

Am = A / n ' calcolo della media dei termini della serie An

Bm = B / n ' calcolo della media dei termini della serie Bn

FOR i = 1 TO n ' governa il calcolo delle differenze alle medie , le somme progressive dei prodotti,        &
```

```
            ' le somme progressive dei quadrati
Ao = ( A( i ) - Am )   ' calcolo differenze alla media per A(i)
Bo = ( B( i ) - Bm )   ' calcolo differenze alla media per B(i)
SP = SP + ( Ao * Bo )  ' sommatoria progressiva dei prodotti
Aq = Aq + Ao ^ 2  ' sommatoria progressiva dei quadrati Aq
Bq = Bq + Bo ^ 2   ' sommatoria progressiva dei quadrati Bq
NEXT i   ' rimanda alla seconda istruzione FOR i...
RAq = SQR( Aq )  ' radice quadrata della sommatoria progressiva Aq
RBq = SQR( Bq )   ' radice quadrata della sommatoria progressiva Bq
C = ( SP ) / ( RAq * RBq )  ' calcolo finale del coefficiente di correlazione
DsA =  RAq / SQR( n  )   ' calcolo della deviazione standard per la serie An
DsB =  RBq / SQR(  n )   ' calcolo della deviazione standard per la serie Bn
PRINT "C="  ;  C    ' presenta il coefficiente di correlazione
PRINT "Am="  ;  Am    ' presenta la media della serie An
PRINT "Bm="  ;  Bm    ' presenta la media della serie Bn
PRINT "DsA="  ;  DsA   ' presenta il dato della deviazione standard per la serie An
PRINT "DsB="  ;  DsB   ' presenta il dato della deviazione standard  per la serie Bn
```

Esaminato il programma, vediamone l'impiego cercando di evidenziare come i risultati che si ottengono siano rappresentativi dei legami di interdipendenza tra le coppie di serie elaborate. A questo scopo iniziamo a considerare due serie di valori legate tra loro da un semplice rapporto di proporzionalità diretta quale può trovarsi tra il tempo di riempimento di una vasca e il volume d'acqua che in essa si deposita; dopo 7 osservazioni a distanza di 1 ora sia:

An tempo in ore	Bn volume d'acqua mc
1	.75
2	1.5
3	2.25
4	3
5	3.75
6	4.5
7	5.25

con queste due serie di dati procediamo al calcolo degli elementi elaborati dal programma:

```
                F5
n ? 7
A1
? 1
```

..... si introducono tutti i valori fino ad A7 = 7 dopo di che il programma richiede i valori di Bn
B1
?.75
...... si introducono tutti i valori fino a B7 = 5.25 , dopo l'introduzione dell'ultimo si ha la presentazione dei risultati

C = 1
Am = 4
Bm = 3
DsA = 2
DsB = 1.5

L'interpretazione del risultato relativo al coefficiente di correlazione è la seguente:
Il valore di C = 1 ci dice che le due serie di grandezze sono strettamente legate tra loro (massimo valore del coefficiente di correlazione), ciò era atteso dato che volutamente abbiamo ipotizzato un fenomeno fisico in cui l'effetto (volume d'acqua nella vasca) è proporzionale alla causa (tempo trascorso per il versamento).
In questo caso si dice che le due serie di valori sono tra loro **correlate.**
Un secondo esempio significativo si può ottenere ipotizzando che la vasca, relativa all'esercizio precedente, in cui sono stati immessi 5.25 mc d'acqua, venga svuotata completamente con regolarità con la stessa portata di immissione. Si rilevano le due serie di valori di questa nuova operazione:

An tempo in ore	Bn volume d'acqua mc
1	4.5
2	3.75
3	3
4	2.25
5	1.5
6	.75
7	0

Se i valori delle due serie vengono introdotti nel programma abbiamo:

F5
n ? 7
A1
? 1
...... si introducono i termini fino ad A7=7
B1
? 4.5
...... si introducono i termini fino a B7= 0 e si ottiene
C = -1
Am = 4
Bm = 2.25
DsA = 2
DsB = 1.5

Il valore di C = -1 ci dice che le due serie di grandezze sono strettamente legate tra loro (massimo valore del coefficiente di correlazione negativo), ciò era atteso dato che volutamente abbiamo ipotizzato un fenomeno fisico in cui l'effetto (volume d'acqua nella vasca) è dipendente dalla causa in senso decrescente (tempo trascorso per lo svuotamento).
In questo caso si dice che le due serie di valori sono **inversocorrelate.**

Un terzo esempio aiuterà a meglio comprendere l'utilità dell'algoritmo di correlazione; supponiamo che l'incremento del volume dell'acqua nella vasca non sia regolare, sia a causa di immissione discontinua, sia a causa di perdite nella vasca, e che i rilievi fatti portino a due nuove serie di valori come sotto riportato:

An tempo in ore	Bn volume d'acqua mc
1	1
2	.8
3	1.2
4	2
5	4
6	3
7	8

si avrà:

```
                F5
n ? 7
A1
? 1
.... fino ad A 7 =7
B1
? 1
... fino a B7 = 8


C= .8540594
Am = 4
Bm = 2.857143
DsA = 2
DsB = 2.358484
```

Il nuovo valore di C = .8540594 ci dice che le due serie di grandezze non sono strettamente legate tra loro (il valore del coefficiente di correlazione è sensibilmente inferiore ad 1), ciò dipende dal fatto che elementi di perturbazione hanno inciso sul fenomeno in esame riducendo il rapporto di proporzionalità che era caratteristico del primo esercizio.
E' utile mostrare in quali casi il valore del coefficiente di correlazione è nullo o molto piccolo; per fare ciò si dovrebbero confrontare due serie molto numerose di valori presi casualmente, ad esempio tirando 100 volte due coppie di dadi diversamente colorati per associare al colore di una coppia i valori della serie An e al colore dell'altra coppia i valori della serie Bn.
Dato che questa operazione è tediosa si possono costruire due serie di numeri ad arte tali da ottenere che il coefficiente di correlazione tra di esse sia nullo così come mostrano le serie di valori delle seguenti tabelle:

An	Bn
4	-4
-6	-6
-2	2
4	-4

che inserite a programma permettono di ottenere:

F5

n ? 4
A1
? 4
..... fino ad A4 = 4
B1
? - 4
..... fino a B4 = - 4

C = 0
Am = 0
Bm = -3
DsA = 4.24264
DsB = 3

come era nelle intenzioni le due serie di valori An; Bn portano a C = 0, ciò denuncia la completa indipendenza di una serie rispetto all'altra. In questo caso le due serie di valori si dicono **scorrelate.**

9.2 La generazione di serie di numeri casuali

Per diversi impieghi, tra i quali le esercitazioni a scopo didattico con gli algoritmi di correlazione, sono disponibili in Qbasic alcune istruzioni che gestiscono la generazione automatica di serie di numeri casuali, tali quindi da essere, se prese con un numero sufficiente di termini, praticamente scorrelate .
Le istruzioni in oggetto sono:

RANDOMIZE TIMER
RND

L'istruzione RANDOMIZE TIMER ha il compito di. inizializzare il generatore di numeri casuali che è implementato in Qbasic.
L'istruzione RND restituisce i numeri casuali inizializzati dall'istruzione precedente, i numeri resi sono compresi tra 0 ed 1.
Vediamo come compilare un piccolo programma per la generazione di una serie di numeri casuali:

```
CLS ' pulisci lo schermo

INPUT "n" ; n ' richiesta del numero dei termini della serie di valori casuali

RANDOMIZE TIMER ' inizializza il generatore di numeri casuali

FOR i =1 TO n ' imposta la restituzione di n numeri casuali da parte dell'istruzione RND

y = RND ' restituisce gli n numeri casuali ponendoli uguali ad y

PRINT y ' presenta gli n numeri casuali

NEXT i ' rimanda all'istruzione FOR i ... per la restituzione dei successivi numeri casuali
```

Se desideriamo ad esempio una serie di 12 numeri casuali premiamo F5 e si ha:
n? 12
.
. e di seguito la sequenza di 12 numeri diversi tra loro che non si elencano dato
. che non si ritroverebbero mai uguali nel provare il programma.

E' perciò immediata la conseguenza che con due giri di programma, per lo stesso valore di n, si hanno le presentazioni di due serie di numeri diverse tra loro.
Dato che i numeri generati sono compresi tra 0 ed 1, se si vogliono valori tra 0 e 10, 0 e 100, ecc.. è necessario modificare l'istruzione y = RND in:

y = 10 * RND; y = 100 * RND; ecc...

Se si desiderano serie di numeri casuali interi si deve impiegare l'istruzione **INT**, che opportunamente inserita consente di restituire la sola parte intera di un numero:

y = INT (10 * RND); y = INT (100 * RND); . .ecc..

9.3 L'impiego del generatore di numeri casuali nei processi di correlazione

Disponendo del "generatore di numeri casuali" implementato in Qbasic è interessante vedere il comportamento dei coefficienti di correlazione quando nel programma del paragrafo 9.1 si inserisce la routine illustrata nel paragrafo 9.2.
Vediamo anzitutto come il generatore può essere inserito nel contesto del programma di calcolo citato:
Il generatore di numeri casuali prende il posto del sistema di caricamento delle memorie volatili che normalmente immagazzinano, dopo digitazione, gli n valori delle serie An e Bn; infatti per questo tipo di esercitazioni le serie An e Bn sono prodotte dal generatore stesso.
Il programma modificato in questo senso è qui compilato e commentato:

```
CLS ' pulisce lo schermo

INPUT "n=" ; n 'entra il numero dei termini delle due serie (massimo valore di n =255)

DIM A( n ), B( n ) ' dimensionamento matrici volatili per contenere i valori delle due serie di dati
                   ' prodotte dal generatore di numeri casuali

RANDOMIZE TIMER ' inizializza il generatore di numeri casuali

FOR i =1 TO n ' imposta la restituzione di numeri casuali da parte dell'istruzione RND
              ' per caricarli nelle memorie volatili A(n) ;B(n)

A(i) = RND ' restituisce n numeri casuali ponendoli nella matrice A(n)

B(i) = RND ' restituisce n numeri casuali , diversi dai precedenti, ponendoli nella matrice B(n)

NEXT i ' rimanda all'istruzione FOR i ... per la restituzione dei successivi numeri casuali

FOR i = 1 TO n ' governa il calcolo delle sommatorie progressive per il

A = A + A( i ) ' successivo calcolo delle medie di An e Bn

B = B + B( i )

NEXT i ' rimanda a FOR i... per il completamento delle sommatorie

Am = A / n ' calcolo della media dei termini della serie An

Bm = B / n ' calcolo della media dei termini della serie Bn

FOR i = 1 TO n ' governa il calcolo delle differenze alle medie , le somme progressive dei prodotti, &
```

```
                    ' le somme progressive dei quadrati
Ao = ( A( i ) - Am )  ' calcolo differenze alla media per A(i)
Bo = ( B( i ) - Bm )  ' calcolo differenze alla media per B(i)
SP = SP + ( Ao * Bo ) ' sommatoria progressiva dei prodotti
Aq = Aq + Ao ^ 2  ' sommatoria progressiva dei quadrati Aq
Bq = Bq + Bo ^ 2   ' sommatoria progressiva dei quadrati Bq
NEXT i   ' rimanda alla seconda istruzione FOR i...
RAq = SQR( Aq )  ' radice quadrata della sommatoria progressiva Aq
RBq = SQR( Bq )  ' radice quadrata della sommatoria progressiva Bq
C = ( SP ) / ( RAq * RBq )  ' calcolo finale del coefficiente di correlazione
DsA =  RAq / SQR( n )  ' calcolo della deviazione standard per la serie An
DsB =  RBq / SQR( n )  ' calcolo della deviazione standard per la serie Bn
PRINT "C=" ;  C  ' presenta il coefficiente di correlazione
PRINT "Am=" ;  Am  ' presenta la media della serie An
PRINT "Bm=" ;  Bm   ' presenta la media della serie Bn
PRINT "DsA=" ;  DsA  ' presenta il dato della deviazione standard per la serie An
PRINT "DsB=" ;  DsB  ' presenta il dato della deviazione standard  per la serie Bn
```

L'impiego di questo programma sperimentale è utile perché permette di constatare come varia il coefficiente di correlazione in dipendenza del numero dei termini che formano le due serie di numeri casuali. Infatti per pochi termini, anche se generati in modo casuale, si ha ancora correlazione tra le due serie, aumentando il numero dei termini delle serie si vede che il valore di C decresce fino ad indicare livelli di scorrelazione sensibili.

Per verificare quanto detto facciamo girare il programma per diversi valori di n e prendiamo nota di come varia il corrispondente valore di C; questa operazione viene svolta in modo automatico senza che l'operatore possa rendersi conto dei valori che compongono le due serie messe a calcolo di volta in volta.

I risultati di questo esercizio non possono essere riportati numericamente nel testo dato che non sono mai ripetibili a seguito della variabilità casuale dei numeri delle due serie che incide naturalmente sul coefficiente di correlazione. L'andamento dei risultati, di volta in volta diversi, può essere però commentato con profitto:

La prima osservazione riguarda serie di 2 o 3 termini, per un numero così limitato di valori il coefficiente di correlazione C è naturalmente molto elevato e può oscillare <u>casualmente</u> intorno a +1 ed a -1. E' poco probabile infatti che pochi numeri presi a caso possano mostrare caratteristiche di scorrelazione quali quelle studiate appositamente per l'ultimo esercizio del paragrafo 9.1.

Per serie da 4 a 50 termini si nota che C decresce sensibilmente oscillando tra valori ora positivi ora negativi, compresi <u>indicativamente</u> in una fascia che varia da .70 a .10. Ciò dimostra che le serie iniziano a manifestare segni evidenti di scorrelazione.

Per serie da 50 a 250 termini si osservano molti casi di scorrelazione quasi totale con valori di C dell'ordine di + / - .001, si presentano ancora alcuni valori di C piazzati attorno a +/- .1.
L'esercizio svolto ha dato un'idea di come vengano trattate le serie numeriche mediante gli algoritmi di correlazione, la notevole variabilità dei risultati dei calcoli dipende soltanto dalle caratteristiche della distribuzione casuale dei valori numerici delle serie.

9.4 La correlazione tra funzioni di tabella (matrice)

E' significativa la ricerca dei coefficienti di correlazione tra funzioni di tabella che, diversamente dalle serie numeriche, devono essere confrontate mediante la **correlazione di posizione.**
La correlazione di posizione è un procedimento di calcolo uguale a quello mostrato nel paragrafo 9.1 che viene ripetuto tante volte quanti sono gli elementi della matrice, in dipendenza della posizione degli elementi di una matrice rispetto agli elementi dell'altra tenuti fissi.
Per comprendere meglio quando detto vediamo un semplice esempio che interessa due funzioni di matrice espresse come segue:

Funzione A(x)	Funzione B(x)
10	7
2	9
15	6
8	5

La prima operazione di correlazione si esegue, mediante il programma illustrato nel paragrafo 9.1, sulle due funzioni di matrice come se fossero due serie di numeri e si ricava il coefficiente di correlazione di C1 = -.6624057.
Se si spostano gli elementi della funzione B(x) di un posto in modo che risulti:

Funzione A(x)	Funzione B(x)
10	9
2	6
15	5
8	7

e con questa nuova disposizione si esegue il calcolo del secondo coefficiente di correlazione.
Si ha: C2 = -.1179627.
Se si spostano ancora di un posto gli elementi della funzione B(x) si ha

Funzione A (x)	Funzione B(x)
10	6
2	5
15	7
8	9

ed il valore del terzo coefficiente di correlazione è C3 = .3901842.
Infine l'ultimo spostamento degli elementi della funzione B(x):

Funzione A(x)	Funzione B(x)
10	5
2	7
15	9
8	6

ed il valore del quarto coefficiente di correlazione è C4 = .3901842.
Con questo procedimento abbiamo calcolato, mediante quattro spostamenti, 4 valori di C che rappresentano a loro volta una funzione di tabella:

variabile di spostamento S	Funzione C(s)
0	-.6624057
1	-.1179627
2	.3901842
3	.3901842

pertanto è più corretto, in questi casi, sostituire la dizione coefficiente di correlazione con la dizione **funzione di correlazione di matrice.**
La funzione di correlazione di matrice esprime come varia il legame di interdipendenza tra due funzioni di tabella, il numero più elevato in valore assoluto che risulta dai calcoli di C(x) è il massimo di correlazione, o di inversocorrelazione, esistente tra le due funzioni; nel nostro esempio si ha per C1= -.6624057 il massimo di inversocorrelazione.
Generalmente la ricerca del massimo della funzione di correlazione di matrice viene eseguita tra funzioni di tabella costituite da molti elementi perciò risulta non praticabile il metodo di calcolo adottato nell'esempio che ne ha soltanto 4. Un interessante programma implementato in Qbasic permette il calcolo automatico per funzioni di matrice con un massimo di 127 elementi; il programma viene ora impiegato ,come esercizio, su matrici a 16 termini.
Le dimensioni della matrice volatile che sarà distinta dalla sigla B1 dipendono dal numero degli elementi, tra ordinari e spostati, pari al prodotto $n \cdot (n - 1) = 16 \cdot (16 - 1) = 240$.
Prima di compilare il programma di calcolo della funzione di correlazione di matrice è opportuno anticiparne la struttura:

1° - Le prime 10 istruzioni che seguono **CLS** sono dedicate al caricamento manuale, da parte dell'operatore, degli elementi che costituiscono le due matrici da correlare A(n) e B(n), per un massimo di 16 elementi.
2° - Con l'istruzione **FOR k = 0 TO (n - 1)**, per ciascuno degli n valori di k, si comanda l'esecuzione di uno degli n processi di correlazione tra le due funzioni di matrice.
3° - Con le due istruzioni **FOR x = 1 TO n** si comandano le n operazioni che portano al computo del primo valore C1
4°- Con le istruzioni
y = x + k
IF y > n THEN y1 = (y - n) ELSE y1 = y
si costruisce la variabile indipendente y1 che deve gestire il caricamento degli (n · (n - 1)) elementi della matrice B1(x) derivata dalla B(n) dopo (n - 1) spostamenti così come mostrato nell'esempio dell'inizio paragrafo.

Il procedimento è spiegato con il seguente esempio svolto per n = 3:
inizia la prima scansione dei 3 elementi di A(n) e B(n) eseguita per k = 0 ed x variabile da 1 ad 3 e con y1 = y = 1; 2; 3, la scansione in x legge nell'ordine gli elementi A(1);A(2);A(3), la scansione in y1 legge nell'ordine gli elementi B(1);B(2);B(3), raggiunto n = 3 il valore di k diventa 1 (vedi istruzione al punto 2°) e tutti i valori di y = x + k sono incrementati di 1; y1 = y = 2; 3; 4 ciò provoca uno spostamento di un posto nella lettura di B(n), quando y raggiunge il valore 4 > n = 3 viene posto y1 = y - n = 4 - 3 = 1 e la scansione legge gli elementi nell'ordine A(1);A(2);A(3); B(2);B(3);B(1). Il processo si ripete per k = 2 e tutti i valori di y = x + k sono incrementati di 2; y1 = y = 3; 4; 5 ciò provoca uno secondo spostamento di un posto nella lettura di B(n), quando y raggiunge il valore 4 > n = 3 il valore di y1 viene prima posto: y1= y - n = 4 - 3 = 1 e dopo per 5 > n = 3 viene posto: y1 = y - n = 5 - 3 = 2 e la scansione legge gli elementi nell'ordine A(1); A(2); A(3); B(3); B(1); B(2).
5° - Con l'istruzione B1(x) = B(y1) si comanda il caricamento della matrice B1(x), ad (n · (n-1)) elementi, con i valori ricavati tramite spostamenti dalla B(n).
6° - Seguono le istruzioni di calcolo del coefficiente di correlazione già commentate nel programma illustrato al paragrafo 9.1.
7°- Il programma di calcolo progressivo si completa con le due istruzioni NEXT x.
8°- Il programma generale si chiude con l'istruzione NEXT k che rimanda la routine all'inizio per iniziare il calcolo di un successivo valore di C.

Il programma citato è ora compilato e commentato, esso non prevede la presentazione dei calcoli relativi alle medie ed alle deviazioni standard che se necessario possono essere aggiunte dal lettore.

```
CLS ' pulisce lo schermo

INPUT "n" ; n ' entra il numero degli elementi di matrice

DIM A(16 ), B(16), B1(255) ' dimensionamento delle matrici volatili

FOR i = 1 TO n ' governa il caricamento della matrice A(n)

PRINT "A" ; i ' presenta il simbolo A(i) per l'istruzione successiva

INPUT A( i ) ' presenta il simbolo ? per l'ingresso valori di A(n)

NEXT i ' rimanda alla prima istruzione FOR i... per il caricamento dei successivi elementi matrice A(n)

FOR i = 1 TO n ' governa il caricamento della matrice B(n)

PRINT "B" ; i ' presenta il simbolo B(i) per l'istruzione successiva

INPUT B( i ) ' presenta il simbolo ? per l'ingresso valori di B(n)

NEXT i ' rimanda alla seconda istruzione FOR i... per il caricamento dei successivi elementi matrice B(n)

FOR k = 0 TO (n - 1) ' istruzione per la formazione di una matrice B1(x)
    ' contiene tutte le possibili configurazioni di posizione degli elementi di B(n) pari ad ( n * ( n-1 ) ) elementi

FOR x = 1 TO n ' esplora la matrice A(n) da 1 ad n per (n-1) volte

y = x + k ' si forma la variabile per la matrice B1(x)

IF y > n THEN y1 = ( y - n ) ELSE y1 = y          &
```

```
B1( x ) = B( y1 )  ' si caricano gli n valori di B(n) disposti in ( n * ( n-1 ) )  posizioni d'ordine diverse in B1(x)

A = A  +  A( x )   ' calcolo medie

B = B  +  B1( x )

NEXT x

Am = A / n    ' calcolo della medie Am dopo le sommatorie

Bm = B / n    ' calcolo della media Bm dopo le sommatorie

FOR  x = 1  TO  n

Ao = ( A( x )  -  Am )   ' calcolo differenze alla media per A(x)

Bo = ( B1( x )  -  Bm )   ' calcolo differenze alla media per B1(x)

SP = SP  +  ( Ao  *  Bo )   ' sommatoria progressiva dei prodotti

Aq = Aq  +  Ao ^ 2   ' sommatoria progressiva dei quadrati Aq

Bq = Bq  +  Bo ^ 2   ' sommatoria progressiva dei quadrati Bq

NEXT x

RAq =  SQR( Aq )   ' radice quadrata della sommatoria progressiva Aq

RBq =  SQR( Bq )   ' radice quadrata della sommatoria progressiva Bq

C =  ( SP ) / ( RAq  *  RBq )   ' calcolo finale degli n coefficienti di correlazione

PRINT "C";  k  ;  "=" ;  C   ' presentazione dei coefficienti di correlazione.

A = 0     '  azzeramento memorie per i successivi calcoli di C
B = 0
Am = 0
Bm = 0
Ao = 0
Bo = 0
SP = 0
Aq = 0
Bq = 0

NEXT k
```

Per provare il programma proponiamoci la determinazione della funzione di correlazione di matrice delle due funzioni di tabella a 10 elementi (n = 10):

variabile	Funzione	Funzione
x	A(x)	B(x)
1	1.7	2.5
2	3.4	2.9
3	5.0	3.2
4	6.4	3.5
5	7.6	3.0
6	8.6	3.7

7	9.3	.6
8	9.8	1.0
9	10.0	1.7
10	11.0	2.0

F5

n? 10
A1
? 1.7
A2
?
3.4
.... di seguito fino a
A10
? 11
B1
? 2.5
B2
? 2.9
..... di seguito fino a
B10
? 2 dopo l'introduzione dell'ultimo elemento di B(x) si ha

C0 = - . 4759656
C1 = - . 5531418
C2 = - . 4766486
C3 = - . 2539985
C4 = . 1339315
C5 = . 4026823
C6 = . 989358
C7 = . 4487832
C8 = 3. 011922 E-02
C9 = -. 2451199

Il calcolo ha portato alla determinazione della funzione di correlazione di matrice che presenta il massimo di correlazione per C6 = .989358, è utile in molte applicazioni evidenziare, sia il valore massimo di correlazione, sia il numero (S) corrispondente allo spostamento effettuato sugli elementi di B(x) per ottenere tale massimo; nel nostro caso abbiamo S = 6.

La funzione di correlazione di matrice viene così espressa:

variabile di spostamento S	C(S)
0	-.4759656
1	-.5531418
2	-.4766486
3	-.2539985
4	.1339315
5	.4026823
6	.989358
7	.4487832
8	3.01192 E-02
9	-.2451199

dove il valore C(0) rappresenta la correlazione esistente tra A(x) e B(x) eseguita senza alcun spostamento degli elementi di B(x),......, dove il valore di C(9) rappresenta la correlazione esistente tra le due funzioni di matrice dopo aver eseguito il nono spostamento degli elementi di B(x). E' di fondamentale importanza rimarcare che le operazioni di correlazioni tra le due funzioni sono eseguite nel presupposto che **A(x) rimanga sempre invariata**, mentre gli spostamenti avvengono soltanto negli elementi di B(x).

9.5 La correlazione tra fenomeni ondulatori casuali

Si definiscono fenomeni ondulatori casuali quelli esprimibili mediante funzioni f(t) che mutano nel tempo sia in ampiezza che in polarità in modo casuale.
Gli algoritmi che abbiamo illustrato nel paragrafo 9.1 riguardano la ricerca dei valori dei coefficienti di correlazione tra serie di grandezze in numero discreto, questi metodi di calcolo non si adattano alla determinazione degli omologhi indicatori della correlazione tra funzioni che esprimono dei fenomeni ondulatori casuali.
Una certa analogia esiste invece tra la correlazione di funzioni di matrice e funzioni ondulatorie casuali, queste ultime possono considerarsi "estensioni" delle prime con un numero infinito di elementi nelle quali lo spostamento (S), definito con un numero discreto di valori, viene sostituito da una variabile di spostamento continua (r).
Gli indicatori del grado di correlazione tra fenomeni ondulatori casuali hanno caratteristiche di vere e proprie funzioni matematiche, per tale motivo sono detti **funzioni di correlazione**.
La teoria sulle funzioni di correlazione è molto vasta e non può essere trattata in questa sede, sede che si propone soltanto di fornire delle informazioni sui metodi di calcolo implementabili in Qbasic.
Le funzioni di correlazione si ottengono per via analitica mediante complesse elaborazioni, i loro algoritmi sono però facilmente implementabile in Qbasic come ora dimostreremo.

9.5.1 La funzione di correlazione tra due fenomeni ondulatori casuali in banda (0 - F)

Se due fenomeni casuali, definiti rispettivamente dalle funzioni del tempo f1(t) e f2(t), hanno tutte le loro componenti ondulatorie contenute in un intervallo di frequenza compreso tra la frequenza zero e la frequenza F, e se in tale intervallo l'ampiezza delle componenti dei due fenomeni è mediamente uniforme, la funzione di correlazione normalizzata tra i fenomeni stessi è data da:

$$C(r) = \frac{\text{Sen}[2\pi F(r - rf)]}{[2\pi F(r - rf)]}$$

dove

C(r) = simbolo della funzione di correlazione

F = frequenza estrema dell'intervallo

r = variabile indipendente di spostamento temporale di f2(t) rispetto a f1(t)

rf = valore iniziale e fisso di spostamento temporale di f1(t) rispetto a f2(t)

Si deve osservare che in alcuni casi può essere rf = 0, ciò significa che le due funzioni sono correlate per r = 0.
Un esempio fisico è utile per inquadrare meglio questa nuova funzione:

Sia f(t) un fenomeno ondulatorio casuale dovuto a vibrazioni meccaniche le cui oscillazioni sono contenute nell'intervallo compreso tra 0 e 1000 Hz . L'azione vibratoria di f(t) si propaghi su due percorsi diversi, percorso 1 e percorso 2, con il percorso 1 superiore al percorso 2, per poi essere percepita nello stesso punto d'ascolto A.
Indicando con f1(t) la vibrazione che si propaga lungo il percorso 1 supponiamo che questo sia coperto in .0005 Sec.
Indicando con f2(t) la vibrazione che si propaga lungo il percorso 2 supponiamo che questo sia coperto in .0003 Sec.
E' evidente che la vibrazione portata in A da f1(t) sarà ritardata rispetto alla vibrazione portata in A da f2(t) del tempo rf = .0005 - .0003 = .0002 Sec.
La funzione di correlazione tra i due fenomeni vibratori presenti nel punto A, funzione che esprime il legame di interdipendenza tra f1(t) e f2(t), sarà pertanto:

$$C(r) = \frac{\text{Sen}\,[\,2\,\pi\,1000\,(\,r - .0002\,)\,]}{[\,2\,\pi\,1000\,(\,r - .0002\,)\,]}$$

A questo punto non resta che computare l'espressione ottenuta, in funzione della variabile di posizione temporale (r), per ottenere l'andamento della funzione di correlazione interessata.
Il computo ed il grafico di C(r) si ottengono dal programma generale che si avvale della corrispondenza simbolica in Qbasic della funzione C(r):

C(r) = SIN (2 * 3.14 * F * (r - rf)) / (2 * 3.14 * F * (r - rf))

che deve essere sviluppata per tutti i valori di r compresi nell'intervallo 0 - ro dove ro >> rf
Il programma compilato e commentato è:

```
' Iniziare digitando la routine per il grafico del reticolo

LINE ( 0 , 160 ) - ( 460 , 160 )  ' comanda il tracciamento dell'asse delle ascisse (asse X)
                                  ' per coordinate a 2 quadranti

LINE ( 0 , 0 ) - ( 0 , 320 )  ' comanda il tracciamento dell'asse delle ordinate ( asse Y)
                              ' per coordinate a 2 quadranti

LOCATE 8,66 : PRINT "F"   ' presentazione simbolo F per istruzione successiva

LOCATE 9,66 :  INPUT F     ' presentazione simbolo  ? per introduzione valore di F

LOCATE 10,66 : PRINT "rf "  ' presentazione simbolo rf  per istruzione successiva

LOCATE 11,66 :  INPUT rf    ' presentazione simbolo  ? per introduzione valore di rf

LOCATE 12,66 :  PRINT "ro"   ' presentazione simbolo ro per istruzione successiva

LOCATE 13,66 : INPUT ro   ' presentazione simbolo  ? per introduzione valore di ro

FOR  r = .0000001  TO  ro  STEP  ( ro / 1000 )  ' impostazione campo variabilità di r  a 1000 passi fissi

C( r ) = SIN ( 2 * 3.14 * F * ( r - rf ) ) / ( 2 * 3.14 * F * ( r - rf ) )  ' calcolo C(r)

PSET( 460 * r / ro , 160 - 160 * C ( r ) ),14  ' tracciamento del grafico andamento C(r) con       &
```

' normalizzazione automatica asse delle ascisse

NEXT r ' rimanda all'istruzione FOR r... per il calcolo dei successivi punti di C(r)

L'esempio fisico illustrato può essere concluso con l'applicazione del programma ora compilato assumendo per ro il valore .002 >> rf; si preme F5 e si ha:

F
? 1000
rf
? .0002
ro
? .002

dopo l'introduzione del valore di ro si ha la presentazione, in giallo, della curva di correlazione riportata in figura 45 in cui l'asse delle ordinate è diviso in 20 intervalli da .1 e l'asse delle ascisse in 20 intervalli da .0001 Sec.
Come avevamo anticipato si constata che il legame di correlazione tra due funzioni è a sua volta una funzione e come tale è rappresentabile graficamente per un qualsiasi numero di valori compatibilmente con il numero dei pixel disponibili sullo schermo.
Dalla curva si osserva:
-La C(r) presenta il suo valore massimo in corrispondenza di r = .0002 Sec. così come era prevedibile dato che tanto è il ritardo che la f1(t) ha rispetto alla f2(t).
-Il profilo della C(r) ha la nota caratteristica delle funzioni Sen x /x.
-La curva di C(r) presenta il primo zero per un valore di
r = ro + 1 / 2F = .0002 + 1 / (2 · 1000) = .0007 Sec.
Ciò indica che per tale valore le due funzioni f1(t) e f2(t) sono completamente scorrelate, cioè il loro legame di interdipendenza è nullo. La cosa può stupire dato che abbiamo ipotizzato che le due vibrazioni provengano dallo stesso fenomeno ondulatorio. La realtà fisica dell'esempio conferma invece questo comportamento che è inoltre mostrato dall'andamento della C(r) che, al di là di altri valori di zero netto, tende a decrescere d'ampiezza indefinitamente per valori di r crescenti indefinitamente.

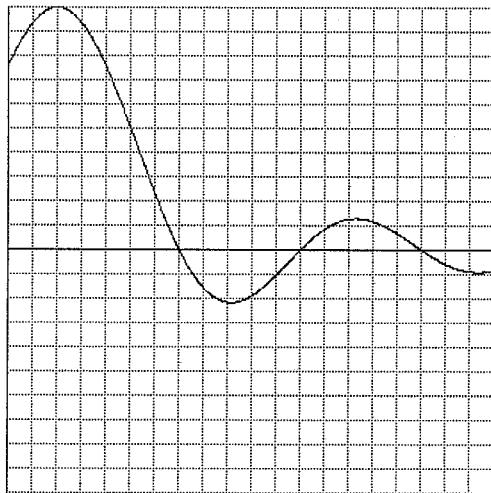



Figura 45
Funzione di correlazione
banda 0 - 1000 Hz
rf = .0002 Sec.

9.5.2 La funzione di correlazione tra due fenomeni ondulatori casuali in banda (F1 - F2)

Se due fenomeni casuali, definiti rispettivamente dalle funzioni del tempo f1(t) e f2(t), hanno tutte le loro componenti ondulatorie contenute in un intervallo di frequenza compreso tra la frequenza F1 e la frequenza F2 e se in tale intervallo l'ampiezza media delle componenti dei due fenomeni è uniforme, la funzione di correlazione normalizzata tra i fenomeni stessi è data da:

$$C(r) = \frac{Sen[2\pi\ DF\ (r - rf)]}{[2\pi\ DF\ (r - rf)]} Cos[2\pi\ Fo\ (r - rf)]$$

dove

C(r) = simbolo della funzione di correlazione
DF = (F2 - F1) / 2
Fo = (F2 + F1) / 2
r = variabile indipendente di spostamento temporale di f2(t) rispetto a f1(t)
rf = valore iniziale e fisso di spostamento temporale di f1(t) rispetto a f2(t)

L ' esempio fisico riportato nel paragrafo 9.5.1 si adatta naturalmente anche a questo tipo di fenomeni casuali per cui possiamo scrivere:
Sia f(t) un fenomeno casuale dovuto a vibrazioni meccaniche le cui oscillazioni sono contenute nell'intervallo compreso tra 5000 Hz e 10000 Hz. Indicando con f1(t) la vibrazione che si propaga lungo il percorso 1 supponiamo che questo sia coperto in .0001 Sec.
Indicando con f2(t) la vibrazione che si propaga lungo il percorso 2 supponiamo che questo sia coperto in .00006 Sec. La vibrazione portata in A da f1(t) sarà ritardata rispetto alla vibrazione portata in A da f2(t) del tempo rf = .0001 - .00006 = .00004 Sec.
Per l'impostazione della funzione di correlazione tra i due fenomeni vibratori presenti nel punto A, funzione che esprime il legame di interdipendenza tra f1(t) e f2(t), si deve anzitutto computare:

$$DF = (F2 - F1) / 2 = (10000 - 5000) / 2 = 2500\ Hz$$

$$Fo = (F2 + F1) / 2 = (10000 + 5000) / 2 = 7500\ Hz$$

da cui si ha

$$C(r) = \frac{Sen[2\pi\ 2500\ (r - .00004)]}{[2\pi\ 2500\ (r - .00004)]} Cos[2\pi\ 7500\ (r - .00004)]$$

Il calcolo ed il grafico di C(r) si ottengono dal programma generale che si avvale della corrispondenza simbolica in Qbasic della funzione C(r) spezzata per comodità di compilazione in due parti:

```
Y1= SIN ( 2 * 3.14 * ( ( F2 - F1 ) / 2 ) * ( r - rf ) ) / ( 2 * 3.14 * ( ( F2 - F1 ) / 2) * ( r - rf ) )

Y2 = COS ( 2 * 3.14 * ( ( F2 + F1 ) / 2 ) * ( r - rf ) )

C( r ) = Y1 * Y2
```

che deve essere sviluppata per tutti i valori di r compresi nell'intervallo 0-ro dove ro >> rf
Il programma compilato e commentato è:

```
'Iniziare digitando la routine per il grafico del reticolo

LINE (0 , 160 ) - ( 460 , 160 ) ' comanda il tracciamento dell'asse delle ascisse (asse X)
                                ' per coordinate a 2 quadranti
LINE ( 0 , 0 ) - ( 0 , 320 )  ' comanda il tracciamento dell'asse delle ordinate ( asse Y)
                              ' per coordinate a 2 quadranti

LOCATE 8,66 : PRINT "F1"  ' presentazione simbolo F1 per istruzione successiva

LOCATE 9,66 : INPUT F1     ' presentazione simbolo ? per introduzione valore di F1

LOCATE 10,66 : PRINT "F2"  ' presentazione simbolo F2 per istruzione successiva

LOCATE 11,66 : INPUT F2  ' presentazione simbolo ? per introduzione valore di F2

LOCATE 12,66 : PRINT "rf " ' presentazione simbolo rf per istruzione successiva

LOCATE 13,66 : INPUT rf   ' presentazione simbolo ? per introduzione valore di rf

LOCATE 14,66 : PRINT "ro"  ' presentazione simbolo ro per istruzione successiva

LOCATE 15,66 : INPUT ro   ' presentazione simbolo ? per introduzione valore di ro

FOR r = .0000001 TO ro STEP ( ro / 1000 ) ' impostazione campo variabilità di r a 1000 passi fissi

Y1= SIN ( 2 * 3.14 * ( ( F2 - F1 ) / 2 ) * ( r - rf ) ) / ( 2 * 3.14 * ( ( F2 - F1 ) / 2 ) * ( r - rf ) )

Y2 = COS ( 2 * 3.14 * ( ( F2 + F1 ) / 2 ) * ( r - rf ) )

C( r ) = Y1 * Y2

PSET( 460 * r / ro , 160 - 160 * C ( r ) ),14  ' tracciamento del grafico andamento C(r) con
                                              ' normalizzazione automatica asse delle ascisse

NEXT r ' rimanda all'istruzione FOR r... per il calcolo dei successivi punti di C(r)
```

L'esempio può essere concluso con l'applicazione del programma ora compilato assumendo per ro il valore .0004 >> rf:

```
F5
                         F1
                         ? 5000
                         F2
                         ? 10000
                         rf
                         ? . 00004
                         ro
                         ? . 0004
```

dopo l'introduzione del valore di ro si ha la presentazione, in giallo, della curva di correlazione riportata in figura 46 in cui l'asse delle ordinate è diviso in 20 intervalli da .1 e l'asse delle ascisse in 20 intervalli da .00002 Sec.

Dalla curva si osserva:

-La C(r) presenta il suo valore massimo in corrispondenza di r = .00004 Sec., così come era prevedibile dato che tanto è il ritardo che la f1(t) ha rispetto alla f2(t).
-Il profilo della C(r) ondula secondo la funzione coseno modulata in Sen x/x, infatti la funzione di correlazione è il prodotto di due funzioni: una cosinusoidale, che varia rapidamente con il variare di r, e una del tipo Sen x/x, che varia, con il variare di r, meno rapidamente della prima.
-Il legame di correlazione di questo tipo di fenomeni dipende prevalentemente dalla larghezza della banda delle frequenze che li compongono e relativamente dal valore di r.

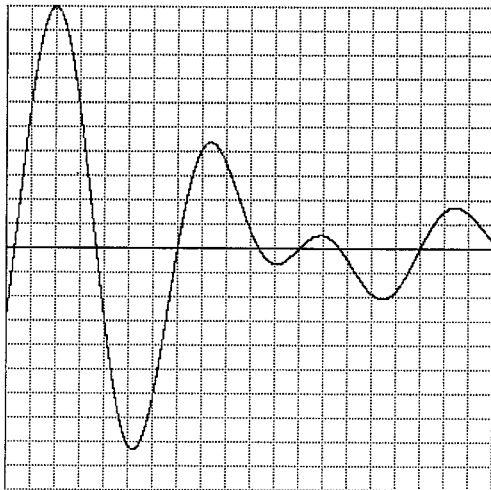



Figura 46
Funzione di correlazione
in banda 5000 - 10000 Hz
rf = .00004 Sec.

9.6 Le funzioni di correlazione per fenomeni casuali a due stati

Per concludere questo capitolo è necessario accennare a due utilissimi algoritmi di correlazione che giuocano un ruolo fondamentale nell'analisi dei fenomeni casuali.
I fenomeni ondulatori casuali, dei tipi già trattati in precedenza, sono definiti da una funzione f(t) che muta nel tempo in ampiezza e polarità in modo casuale.
Un fenomeno casuale a due stati è definito invece da una funzione f (t) che ha sempre ampiezza costante e varia soltanto in polarità in modo casuale.

9.6.1 La funzione di correlazione tra fenomeni casuali a due stati in banda (0 - F)

Se due fenomeni casuali a due stati, definiti rispettivamente dalle funzioni del tempo f1(t) e f2(t), hanno tutte le loro componenti frequenziali contenute in un intervallo di frequenza compreso tra la frequenza zero e la frequenza F, la funzione di correlazione normalizzata tra i fenomeni stessi è data da:

$$Cl(r) = (2/\pi)\ \text{Arcsen} \frac{\text{Sen}[2\pi F(r-rf)]}{[2\pi F(r-rf)]}$$

dove

Cl(r) = simbolo della funzione di correlazione tra funzioni casuali a due stati
F = frequenza estrema dell'intervallo
r = variabile indipendente di spostamento temporale di f2(t) rispetto a f1(t)

rf = valore iniziale e fisso di spostamento temporale di f1(t) rispetto a f2(t)

Il computo ed il grafico di Cl(r) si ottengono dal programma generale che si avvale della corrispondenza simbolica in Qbasic della funzione Cl(r) che per semplificare la compilazione e scritta con due istruzioni:

Y = SIN (2 * 3.14 * F * (r - rf)) / (2 * 3.14 * F * (r - rf))

Cl(r) = (2 / 3.14) * ATN (Y / SQR (-Y * Y + 1))

che deve essere sviluppata per tutti i valori di r compresi nell'intervallo 0-ro dove ro >> rf.
Il programma compilato e commentato è il seguente:

```
' Iniziare digitando la routine per il grafico del reticolo

LINE ( 0 , 160 ) - ( 460 , 160 ) ' comanda il tracciamento dell'asse delle ascisse (asse X)
                                 ' per coordinate a 2 quadranti

LINE ( 0 , 0 ) - ( 0 , 320 )  ' comanda il tracciamento dell'asse delle ordinate ( asse Y)
                              ' per coordinate a 2 quadranti

LOCATE 8,66 : PRINT "F"    ' presentazione simbolo F per istruzione successiva

LOCATE 9,66 : INPUT F    ' presentazione simbolo ? per introduzione valore di F

LOCATE 10,66 : PRINT "rf "  ' presentazione simbolo rf per istruzione successiva

LOCATE 11,66 : INPUT rf    ' presentazione simbolo ? per introduzione valore di rf

LOCATE 12,66 : PRINT "ro"  ' presentazione simbolo ro per istruzione successiva

LOCATE 13,66 : INPUT ro   ' presentazione simbolo ? per introduzione valore di ro

FOR r = .0000001 TO ro STEP ( ro / 1000 ) ' impostazione campo variabilità di r a 1000 passi fissi

Y = SIN ( 2 * 3.14 * F * ( r - rf ) ) / ( 2 * 3.14 * F * ( r - rf ) )  ' calcolo di Cl(r)

Cl( r ) = ( 2 / 3.14 ) * ATN ( Y / SQR ( -Y * Y + 1 ) )          ' in due passi di programma

PSET( 460 * r / ro , 160 - 160 * Cl ( r ) ),14 ' tracciamento del grafico andamento Cl(r) con
                                               ' normalizzazione automatica asse delle ascisse
NEXT r ' rimanda all'istruzione FOR r... per il calcolo dei successivi punti di Cl(r)
```

Se proviamo il programma ora compilato assumendo gli stessi valori utilizzati nell'esercizio del paragrafo 9.5.1 otteniamo

F5

F
? 1000
rf
? .0002
ro
? .002

dopo l'introduzione del valore di ro si ha la presentazione, in giallo, della curva di correlazione riportata in figura 47 in cui l'asse delle ordinate è diviso in 20 intervalli da .1 e l'asse delle ascisse in 20 intervalli da .0001 Sec.

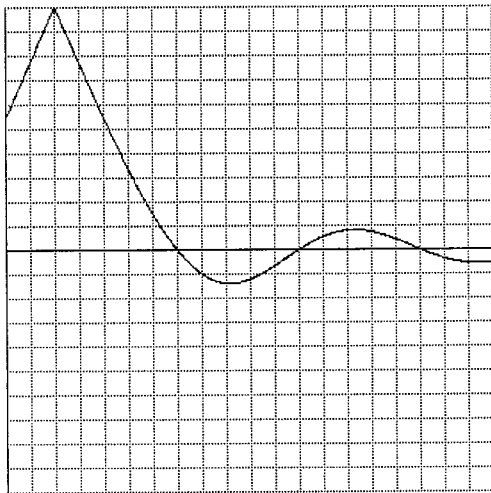



Figura 47
Funzione di correlazione per fenomeni a due stati in banda 0 - 1000 Hz
rf = .0002 Sec.

Dalla curva si osserva:

-Il profilo della Cl(r) mostra una cuspide in corrispondenza del massimo, il restante andamento segue di massima la funzione Sen x/x.
-La Cl(r) presenta il suo valore massimo in corrispondenza di r = .0002 Sec.
-La curva di Cl(r) presenta il primo zero per un valore di r = ro +1/ 2F = .0002 + 1 / (2 ·1000) =
= .0007 Sec.

9.6.2 La funzione di correlazione tra due fenomeni casuali a due stati in banda (F1 - F2).

Se due fenomeni casuali a due stati, definiti rispettivamente dalle funzioni del tempo f1(t) e f2(t), hanno tutte le loro componenti frequenziali contenute in un intervallo di frequenza compreso tra la frequenza F1 e la frequenza F2, la funzione di correlazione normalizzata tra i fenomeni stessi è data da:

$$Cl(r) = (2/\pi)\, \text{Arcsen} \left\{ \frac{\text{Sen}\,[\,2\pi\, DF\,(r - rf)\,]}{[\,2\pi\, DF\,(r - rf)\,]} \text{Cos}\,[\,2\pi\, Fo\,(r - rf)\,] \right\}$$

dove

Cl(r) = simbolo della funzione di correlazione tra funzioni casuali a due stati
DF = (F2 - F1) / 2
Fo = (F2 + F1) / 2
r = variabile indipendente di spostamento temporale di f2(t) rispetto a f1(t)
rf = valore iniziale e fisso di spostamento temporale di f1(t) rispetto a f2(t)

Il computo ed il grafico di Cl(r) si ottengono dal programma generale che si avvale della corrispondenza simbolica in Qbasic della funzione Cl(r) spezzata per comodità di compilazione in quattro parti:

Y1 = SIN (2 * 3.14 * ((F2 - F1) / 2) * (r - rf)) / (2 * 3.14 * ((F2 - F1) / 2) * (r - rf))

Y2 = COS (2 * 3.14 * ((F2 + F1) / 2)* (r - rf))

Y3 = Y1 * Y2

Cl(r) = (2 / 3.14) * ATN (Y3 / SQR (-Y3 * Y3 + 1))

che deve essere sviluppata per tutti i valori di r compresi nell'intervallo 0-ro dove ro >> rf
Il programma compilato e commentato è il seguente:

```
' Iniziare digitando la routine per il grafico del reticolo

LINE ( 0 , 160 ) - ( 460 , 160 )  ' comanda il tracciamento dell'asse delle ascisse (asse X)
                                  ' per coordinate a 2 quadranti

LINE ( 0 , 0 ) - ( 0 , 320 )  ' comanda il tracciamento dell'asse delle ordinate ( asse Y)
                              ' per coordinate a 2 quadranti

LOCATE 8,66 : PRINT "F1"  ' presentazione simbolo F1 per istruzione successiva

LOCATE 9,66 : INPUT F1  ' presentazione simbolo ? per introduzione valore di F1

LOCATE 10,66 : PRINT "F2"  ' presentazione simbolo F2 per istruzione successiva

LOCATE 11,66 : INPUT F2  ' presentazione simbolo ? per introduzione valore di F2

LOCATE 12,66 : PRINT "rf "  ' presentazione simbolo rf per istruzione successiva

LOCATE 13,66 : INPUT rf  ' presentazione simbolo ? per introduzione valore di rf

LOCATE 14,66 : PRINT "ro"  ' presentazione simbolo ro per istruzione successiva

LOCATE 15,66 : INPUT ro  ' presentazione simbolo ? per introduzione valore di ro

FOR r = .0000001 TO ro STEP ( ro / 1000 ) ' impostazione campo variabilità di r a 1000 passi fissi  &
```

```
Y1= SIN ( 2 * 3.14 * ( ( F2 - F1 ) / 2) * ( r - rf ) ) / ( 2 * 3.14 * ( ( F2 - F1 ) / 2) * ( r - rf ) )

Y2 = COS ( 2 * 3.14 * ( ( F2 + F1 ) / 2 ) * ( r - rf ) )  ' calcolo della Cl(r) in 4 passi di programma

Y3 = Y1 * Y2

Cl (r ) = ( 2 / 3.14 ) * ATN ( Y3 / SQR ( -Y3 * Y3 + 1 ) )

PSET( 460 * r / ro , 160 - 160 * Cl( r ) ),14  ' tracciamento del grafico andamento Cl(r) con
                                               ' normalizzazione automatica asse delle ascisse

NEXT r  ' rimanda all'istruzione FOR r... per il calcolo dei successivi punti di Cl(r)
```

Si prova il programma con i dati impiegati nell'esercizio del paragrafo 9.5.2

```
F5
                    F1
                    ? 5000
                    F2
                    ? 10000
                    rf
                    ? .00004
                    ro
                    ? .0004
```

dopo l'introduzione del valore di ro si ha la presentazione della curva di figura 48

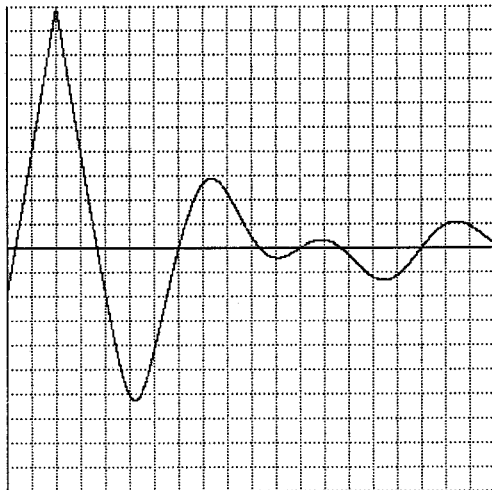



Figura 48
Funzione di correlazione per fenomeni a due stati banda 5000 - 10000 Hz rf = .00004 Sec.

Dalla figura si osserva:
-l'asse delle ordinate è diviso in 20 intervalli da .1
-l'asse delle ascisse è diviso in 20 intervalli da .00002 Sec.
-Il profilo della Cl(r) presenta una cuspide sul massimo, lontano dal massimo la curva ondula secondo la funzione coseno modulata in Sen x/x .
-La Cl(r) presenta il suo valore massimo in corrispondenza di r = .00004 Sec.

-Il legame di correlazione di questo tipo di fenomeni dipende prevalentemente dalla larghezza della banda delle frequenze che li compongono e relativamente dal valore di r.

CAPITOLO 10
I POLINOMI DI BUTTERWORTH E DI CHEBYCHEV E LE LORO TRASFORMAZIONI

In molti studi di carattere tecnico vengono utilizzati alcuni algoritmi che vanno sotto il nome di polinomi di Butterworth e di Chebychev.
Questi algoritmi, grazie alle loro doti di trasformabilità, si possono impiegare come strutture di mascheramento, strutture in grado di bloccare lo sviluppo di altre funzioni in un qualsiasi intervallo stabilito dell'asse delle ascisse.
I polinomi sono funzioni parametriche implementabili in Qbasic al fine di consentirne il tracciamento degli andamenti, sia per la valutazione della scelta fatta sui parametri, sia per l'eventuale aggiustaggio degli stessi onde ottenere i profili desiderati delle curve .

10.1 Il polinomio di Butterworth

La caratteristica del profilo del polinomio di Butterworth è dovuta al particolare comportamento della sua funzione che forma una sorta di gradino a scendere, la parte alta della pedata inizia da $x = 0$ e resta elevata per un tratto, per poi decrescere rapidamente, da un certo valore di x in poi, a livelli molto piccoli.
Il polinomio di Butterwort è espresso mediante la funzione parametrica normalizzata:

$$y = \frac{1}{\left[\, 1 + (x \,/\, xt)^{2n} \,\right]^{1/2}}$$

in cui

x = variabile indipendente

xt = parametro di taglio del polinomio

n = ordine del polinomio o parametro di pendenza

Nell'intervallo compreso tra $x = 0$ e $x << xt$ si ha $y = 1$.
Il valore del parametro xt stabilisce l'ascissa per la quale la funzione si riduce dal valore massimo 1 al valore .707.
Più è elevato il valore di xt più si allontana il tratto decrescente della funzione dall'origine degli assi.
Più è piccolo il valore di xt più si avvicina il tratto decrescente della funzione all'origine degli assi.
Il valore del parametro n governa la pendenza del tratto di curva decrescente.
Più è elevato il valore di n maggiore è la pendenza del tratto discendente della funzione.
Più è piccolo il valore di n minore è la pendenza del tratto discendente della funzione.
Come si comprende dalle definizioni dei parametri, l'andamento del polinomio di Butterworth è totalmente dipendente da questi che possono essere opportunamente dimensionati per ottenere la variazione della funzione e il corrispondente profilo della curva che la rappresenta, in base alle necessità di impiego.
Nella compilazione del programma dovrà essere prevista l'introduzione del valore massimo (x max) che si desidera assegnare al campo della variabile indipendente.
L'implementazione del polinomio in Qbasic si ottiene facilmente con l'espressione:

y = 1 / (SQR(1 + (x / xt) ^ (2 * n)))

e da questa è immediata la compilazione del programma commentato per il tracciamento del grafico corrispondente:

```
LINE ( 0, 0 )-( 0, 320 )        ' ASSE Y   2 quadranti

LINE ( 0, 160 )-( 460, 160 )  ' ASSE X  2 quadranti

LOCATE 8, 66 : PRINT "xt"   ' indica il parametro xt da introdurre

LOCATE 9, 66 : INPUT xt   ' con il simbolo  ?  chiede il parametro sopra indicato

LOCATE 10, 66 : PRINT "x max"   ' indica il valore massimo della variabile indip. che si vuole presentare

LOCATE 11, 66 : INPUT xm   ' con il simbolo  ?  chiede il valore di x max sopra indicato

LOCATE 12, 66 : PRINT "n"   ' indica il parametro  n  da introdurre

LOCATE 13, 66 : INPUT n   ' con il simbolo  ?  chiede il parametro sopra indicato


FOR  x = 0  TO   xm   STEP  ( xm / 1000 )   ' gestisce il campo ed il passo della variabile indipendente  x

y = 1 / ( SQR( 1 + ( x / xt ) ^ ( 2 * n ) ) )   ' calcola i punti della funzione

PSET ( 460 * x / xm , 160 - 160 * y ), 14 ' presenta la curva della funzione con le ascisse normalizzate

NEXT x   ' rimanda all'istruzione FOR x ... per il calcolo dei punti successivi
```

Iniziamo a lavorare con il programma con l'intento di conoscere l'andamento grafico di un polinomio di Butterworth di ordine n = 5, per xt = 10 e x max = 20

```
F5
                                        xt
                                        ? 10
                                        x  max
                                        ? 20
                                        n
                                        ? 5
```

dopo l'introduzione del valore di n si ha la presentazione della curva mostrata in figura 49.
La figura mostra chiaramente la particolare caratteristica di questo tipo di funzioni; il valore di y resta a livello 1 da x = 0 a x < xt, per x = xt si ha y = .707, per valori di x di poco superiori a xt la funzione decresce rapidamente per poi tendere allo zero per x >> xt.

Se si desidera spostare il tratto pendente della curva verso l'origine degli assi, ad esempio per xt = 7, si ripete la routine di calcolo con il nuovo valore di xt e gli originali valori di n e x max.
Se si vuole invece aumentare la pendenza della curva nel tratto discendente si ripete nuovamente la routine con un nuovo valore di n, ad esempio n = 8, e gli originali valori di xt e x max.
Queste semplici operazioni rendono l'idea della facilità di cambiamento dei parametri del polinomio per adattarlo al meglio alle personali esigenze di lavoro.

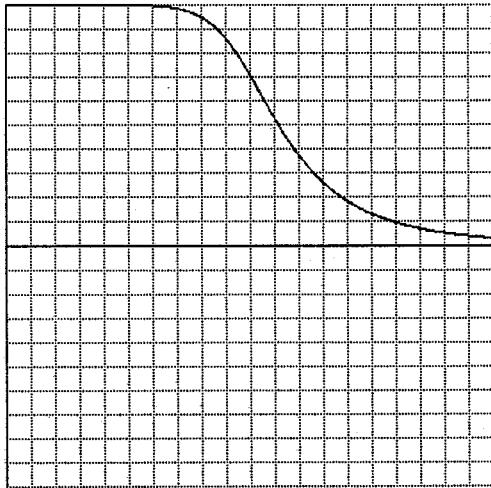



Figura 49
Andamento del polinomio di Butterworth per xt =10, n = 5
Scala asse x = 1/div.
Scala asse y = 1/div.

10.2 La trasformazione simmetrica del polinomio di Butterworth

Il polinomio di Butterworth permette una trasformazione che rovescia l'andamento della funzione rendendo il profilo simmetrico rispetto all'originale.
La caratteristica della trasformazione del polinomio di Butterworth forma una sorta di gradino a salire da livelli molto piccoli di y, che iniziano da x = 0, per poi crescere rapidamente, da un certo valore di x in poi, a livello unitario.
La trasformazione consiste nel cambiare la base della potenza da (x / xt) a (xt / x), si ottiene così una nuova funzione come sotto riportato:

$$y = \frac{1}{\left[1 + (xt / x)^{2n} \right]^{1/2}}$$

in cui

x = variabile indipendente
xt = parametro di taglio del polinomio
n = ordine del polinomio o parametro di pendenza

Nell'intervallo compreso tra x = 0 e x << xt si hanno valori di y molto piccoli.
Il valore del parametro xt stabilisce l'ascissa per la quale la funzione si è incrementata dal valore minimo al valore .707.
Più è elevato il valore di xt più si allontana il tratto crescente della funzione dall'origine degli assi.
Più è piccolo il valore di xt più si avvicina il tratto crescente della funzione all'origine degli assi.

Il valore del parametro n governa la pendenza del tratto di curva crescente.
Più è elevato il valore di n maggiore è la pendenza del tratto ascendente della funzione.
Più è piccolo il valore di n minore è la pendenza del tratto ascendente della funzione.
Questa funzione ha un punto critico per x = 0 dato che in tal caso il rapporto xt/x è infinitamente grande; di ciò si deve tenere conto nella strutturazione del nuovo programma che andremo a compilare .
L'importanza di questa versione del polinomio di Butterworth è pari alla configurazione originale.
Il programma per il tracciamento della funzione è simile al precedente ed è così strutturato:

```
LINE ( 0, 0 )-( 0, 320 )        ' ASSE  Y  2 quadranti

LINE ( 0, 160 )-( 460, 160 ) ' ASSE  X  2 quadranti

LOCATE 8, 66 : PRINT "xt"   ' indica il parametro xt da introdurre

LOCATE 9, 66 : INPUT xt    ' con il simbolo  ? chiede il parametro sopra indicato

LOCATE 10, 66 : PRINT "x max"   ' indica il valore massimo della variabile indip. che si vuole presentare

LOCATE 11, 66 : INPUT xm    ' con il simbolo  ? chiede il valore di x max sopra indicato

LOCATE 12, 66 : PRINT "n"     ' indica il parametro  n  da introdurre

LOCATE 13, 66 : INPUT n    ' con il simbolo  ? chiede il parametro sopra indicato

FOR  x = .0001  TO  xm  STEP  ( xm / 1000 )  ' gestisce il campo ed il passo della variabile
                                              ' indipendente escludendo da esso il valore x=0

y = 1 / ( SQR( 1 + ( xt / x ) ^ ( 2 * n ) ) )  ' calcola i punti della funzione

PSET ( 460 * x / xm , 160 - 160 * y ), 14   ' presenta la curva della funzione con le ascisse normalizzate

NEXT x    ' rimanda all'istruzione FOR x ... per il calcolo dei punti successivi
```

Proviamo il nuovo programma per visualizzare l'andamento grafico della funzione con i dati del polinomio originale: n = 5, per xt = 10 e x max = 20

```
F5

                    xt
                    ? 10
                     x  max
                    ? 20
                    n
                    ? 5
```

dopo l'introduzione del valore di n si ha la presentazione della curva mostrata in figura 50.
Il grafico mostra la curva simmetrica di quella riportata in figura 49; il valore di y resta a livello molto basso per valori di x << xt, per valori di x di poco inferiori a xt la curva cresce rapidamente per giungere al livello y = .707 per x = xt, per valori di x > xt la funzione assume il valore 1.
Qualsiasi cambiamento del profilo della curva è possibile mediante opportuna variazione dei parametri.

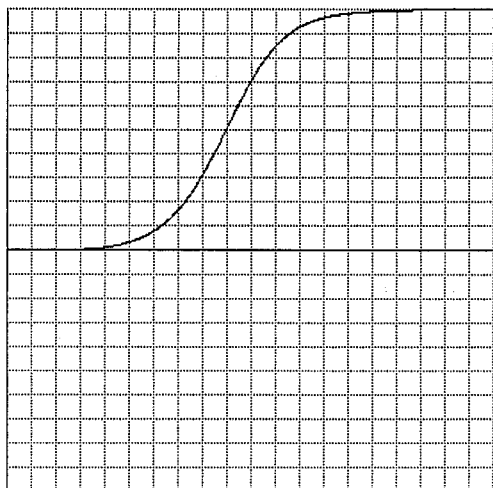



Figura 50
Andamento del polinomio di Butterworth trasformato simmetrico
Scala asse x = 1/div.
Scala asse y = 1/div.

10.3 La doppia trasformazione del polinomio di Butterworth

E' di estremo interesse la trasformazione del polinomio di Butterworth che permette di ottenere una funzione a doppio scalino, a salire per valori di $x < x1$ ed a scendere per valori di $x > x2$, dove x1 ed x2 sono di seguito specificati.
Questa nuova funzione si ottiene dalla trasformazione della base della potenza da (x / xt) in $((x / xo) - (xo / x))$ come è riportato dall'espressione:

$$y = \frac{1}{\left[1 + \left((x / xo) - (xo / x) \right)^{2n} \right]^{1/2}}$$

in cui

x = variabile indipendente

$xo = x1 / .618 = x2 / 1.618$ = parametro di taglio del polinomio

n = ordine del polinomio o parametro di pendenza

Per $x << x1$ si hanno valori di y molto piccoli.
Nell'intervallo compreso tra x1 e x2 si hanno i valori più elevati di y.
Per $x >> x2$ si hanno valori di y molto piccoli.
Il valore del parametro xo stabilisce le due ascisse, x1 ed x2, per le quali la funzione, prima in fase crescente, poi in fase decrescente, assume il valore .707.
Più è elevato il valore di xo più si allontanano i tratti crescenti, decrescenti, della funzione dall'origine degli assi.
Più è piccolo il valore di xo più si avvicinano i tratti crescenti, decrescenti della funzione all'origine degli assi.

Il valore del parametro n governa le pendenze dei tratti di curva .
Più è elevato il valore di n maggiori sono le pendenze dei tratti ascendente, discendente della funzione.
Più è piccolo il valore di n minori sono le pendenze dei tratti ascendente, discendente della funzione.
Questa funzione ha un punto critico per x = 0 dato che in tal caso il rapporto xo/x è infinitamente grande; di ciò si deve tenere conto nella strutturazione del nuovo programma che andremo a compilare .
L'implementazione del polinomio in Qbasic si ottiene con l'espressione:

```
y = 1 / ( SQR( 1 + ( ( xo / x ) - ( x / xo ) ) ^ ( 2 * n ) ) )
```

Il programma per il tracciamento della funzione è così strutturato:

```
LINE ( 0, 0 )-( 0, 320 )        ' ASSE  Y  2  quadranti

LINE ( 0, 160 )-( 460, 160 )  ' ASSE  X  2  quadranti

LOCATE 8, 66 : PRINT "xo"   ' indica il parametro xo da introdurre

LOCATE 9, 66 : INPUT xo   ' con il simbolo  ?  chiede il parametro sopra indicato

LOCATE 10, 66 : PRINT "x max"   ' indica il valore massimo della variabile indip. che si vuole presentare

LOCATE 11, 66 : INPUT xm   ' con il simbolo  ?  chiede il valore di x max sopra indicato

LOCATE 12, 66 : PRINT "n"   ' indica il parametro  n  da introdurre

LOCATE 13, 66 : INPUT n    ' con il simbolo  ?  chiede il parametro sopra indicato

FOR  x = .0001  TO  xm   STEP ( xm / 1000 )  ' gestisce il campo ed il passo della variabile
                                              ' indipendente escludendo da esso il valore x=0

y = 1 / ( SQR( 1 + ( ( xo / x ) - ( x / xo ) ) ^ (2 * n ) ) ) ' calcolo della funzione

PSET ( 460 * x / xm , 160 - 160 * y ), 14   ' presenta la curva della funzione con le ascisse normalizzate

NEXT x   ' rimanda all'istruzione FOR x ... per il calcolo dei punti successivi
```

Proviamo il programma per visualizzare l'andamento grafico della funzione a doppia trasformazione con i valori: n = 6, xo = 5 e x max = 20

```
F5
                              xo
                              ? 5
                               x  max
                              ? 20
                              n
                              ? 6
```

dopo l'introduzione del valore di n si ha la presentazione della curva mostrata in figura 51.
Il grafico mostra la nuova curva del polinomio di Butterworth con doppia trasformazione; il valore di y resta a livello molto basso per valori di x << x1 = .618 xo = 3.09, per valori di x di poco inferiori a x1 = 3.09 la curva cresce rapidamente per giungere al livello y = .707 per

$x = 3.09$, per valori di x compresi tra $x = 3.09$ e $x = x2 = 1.618\ xo = 8.09$ la y mantiene i massimi valori, per $x > x2$ la curva decresce rapidamente, per $x >> x2$ la y ritorna a livelli molto piccoli. Qualsiasi cambiamento del profilo della curva è possibile mediante opportuna variazione dei parametri.

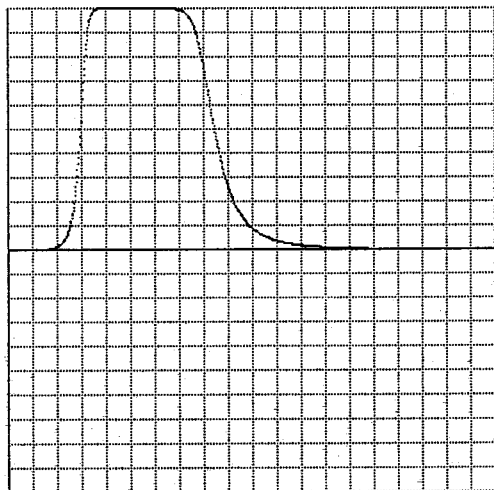



Figura 51
Andamento del polinomio
di Butterworth
doppio trasformato
Scala asse x = 1/div.
Scala asse y = 1/div.

10.4 Sulle caratteristiche di mascheramento del polinomio di Butterworth

Abbiamo accennato che una delle caratteristiche salienti del polinomio di Butterworth è relativa alla proprietà di mascheramento di altre funzioni, mostreremo ora con un esempio tale caratteristica.
Supponiamo di dover presentare la funzione $y1 = \text{Sen}(5x)$ in un intervallo limitato di x compreso tra $x = 0$ e $x = 8$.
Se moltiplichiamo tale funzione per il polinomio di Butterworth descritto nel paragrafo 10.1 abbiamo:

$$Y = \frac{1}{\left[1 + (x / xt)^{2n} \right]^{1/2}} \text{Sen}(5x)$$

In questa nuova funzione la parte polinomiale vale 1 da $x = 0$ a $x < xt$ e assume valori molto piccoli per $x >> xt$ mentre la parte trigonometrica oscilla tra +1 e -1, è chiaro che il prodotto darà modo alla parte trigonometrica di evidenziarsi tra $x = 0$ e $x < xt$ mentre l'attenuerà pesantemente per $x >> xt$.
E' ovvio quindi che se poniamo $xt = 8$ e assegnamo ad n un valore sensibilmente elevato, ad esempio $n = 9$, il prodotto Y risolverà il nostro problema.
L'esempio ha una logica conclusione nella presentazione grafica della funzione prodotto sulla base

dell'implementazione in Qbasic della funzione stessa:

```
Y =  (1 / ( SQR( 1 + ( x / xt )  ^  ( 2 * n ) ) ) ) * SIN ( 5 * x )
```

e del suo inserimento nel programma:

```
LINE ( 0, 0 )-( 0, 320 )        ' ASSE Y  2 quadranti
LINE ( 0, 160 )-( 460, 160 ) ' ASSE  X  2 quadranti
LOCATE 8, 66 : PRINT "xt"  ' indica il parametro xt da introdurre
LOCATE 9, 66 : INPUT xt    ' con il simbolo  ?  chiede il parametro sopra indicato
LOCATE 10, 66 : PRINT "x max"   ' indica il valore massimo della variabile indip. che si vuole  presentare
LOCATE 11, 66 : INPUT xm   ' con il simbolo  ?  chiede il valore di x max sopra indicato
LOCATE 12, 66 : PRINT "n"    ' indica il parametro  n  da introdurre
LOCATE 13, 66 : INPUT n   ' con il simbolo  ?  chiede il parametro sopra indicato
FOR  x = 0   TO  xm    STEP  ( xm / 2000 )  ' gestisce il campo ed il passo della variabile indipendente  x
Y = SIN ( 5 * x ) * ( 1 / ( SQR ( 1 + ( x / xt ) ^ ( 2 * n ) ) ) )  ' calcola i punti della funzione prodotto
PSET ( 460 * x / xm , 160 - 160 * Y ) , 14   ' presenta la curva della funzione con le ascisse normalizzate
NEXT x   ' rimanda all'istruzione FOR x ... per il calcolo dei punti successivi
```

```
F5
                              xt
                              ? 8
                              x  max
                              ? 20
                              n
                              ? 9
```

dopo l'introduzione del valore di n si ha la presentazione della curva mostrata in figura 52 .

La figura mostra che nell'intervallo compreso tra $x = 0$ e $x = 8$ la funzione y = Sen (5 x) oscilla normalmente; per valori di $x > 8$ si ha una brusca attenuazione dell'oscillazione che per $x >> 8$, dati i valori di scala, non è più percepibile.
Esempi analoghi per i polinomi di Butterworth trasformati sono lasciati alla libera scelta del lettore che potrà fare esperienza su questo particolare tema di studio.
La proprietà di mascheramento è utilizzabile, sia per sviluppi di natura analitica, come nell'esempio sopra riportato, assumendo qualsiasi valore di n, sia per la realizzazione di quadripoli fisici per attenuare fenomeni ondulatori non desiderati, in quest'ultimo caso più si eleva il valore di n più risulta complesso il quadripolo.
Nel caso in cui il polinomio di Butterworth definisca le caratteristiche fisiche di un quadripolo, l'azione di mascheramento si esplica in funzione della variabile indipendente x che viene espressa nel dominio della frequenza.
E' nel dominio della frequenza infatti che sono definiti i fenomeni ondulatori che devono essere elaborati.

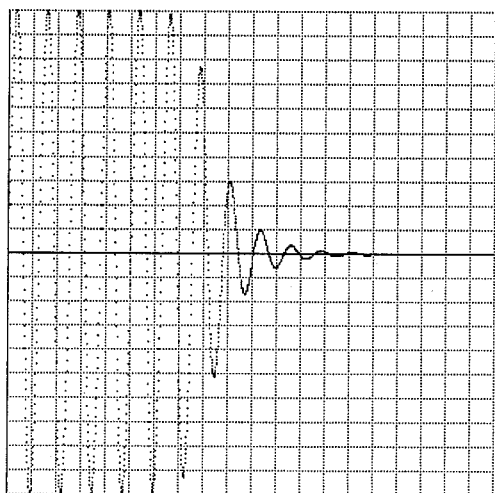



Figura 52
Mascheramento della funzione
Sen 5x con polinomio
di Butterworth
Scala asse x = 1/div.
Scala asse y = 1/div.

10. 5 Il polinomio di Chebyschev

Il polinomio di Chebyschev ha caratteristiche simili al polinomio di Butterworth, permette però di ottenere, a parità di ordine n, pendenze superiori nel tratto decrescente della curva, pagando questo vantaggio con ondulazioni nell'intervallo in cui il secondo polinomio è a livello unitario costante. Il profilo del polinomio di Chebyscev forma una sorta di gradino a scendere in cui la pedata è ondulata, la parte ondulata inizia da x = 0 e resta alta per un certo tratto, per poi decrescere rapidamente, da un certo valore di x in poi, a livelli molto piccoli.
Il polinomio è espresso mediante la funzione parametrica normalizzata:

$$y = \frac{1}{\left[\ 1 + e^2\ C^2n(x)\ \right]^{1/2}}$$

in cui:

e = parametro pendenza e ondulazione

n = ordine del polinomio

Cn(x) è una funzione di n e di x
secondo la tabella:

n	Cn(x)
1	x
2	$2x^2 - 1$
3	$4x^3 - 3x$
4	$8x^4 - 8x^2 + 1$
..	....

La funzione Cn(x) incide sulla pendenza del polinomio e sul numero di ondulazioni presenti nel tratto alto della curva.
Il valore del parametro (e) incide sulla pendenza del tratto discendente del polinomio e sull'ampiezza dell'ondulazione del tratto a livello alto della curva; è particolare il valore e = 1 che provoca ondulazioni che si sviluppano da y = 1 ad y = .707. Valori di e < 1 portano a ondulazioni inferiori a .707 e pendenze ridotte, valori di e > 1 portano ad ondulazioni superiori a .707 e pendenze elevate. Come si vede il polinomio di Chebyschev è definito da una espressione che risulta tanto più complicata quanto è più elevato il valore di n; questo fatto consiglia la compilazione del programma di calcolo e presentazione in modo che si possa scegliere una delle quattro funzioni Cn(x) sopra riportate. Per funzioni Cn(x) di ordine superiore al 4 è lasciata come esercizio al lettore la stesura ed implementazione in Qbasic.

In base alle corrispondenze simboliche in Qbasic si ha:

Per la funzione Cn(x) posta uguale a Yn

n = 1; C1(x) = y1 = x

n = 2; C2(x) = y2 = 2 * (x ^ 2) - 1

n = 3; C3(x) = y3 = 4 * (x ^ 3) - 3 * x

n = 4; C4(x) = y4 = 8 * (x ^ 4) - 8 * (x ^ 2) + 1

Per la selezione del Cn(x) voluto, per n da 1 a 4, si compila la somma S delle quattro funzioni moltiplicate ciascuna per un diverso coefficiente, da k1 a k4, il programma, in base al valore assegnato ad n, pone un solo valore di k = 1 abilitando la corrispondente funzione, le altre tre vengono moltiplicate per k = 0 e non incidono sulla somma S.
Si ha per S l'espressione:

S = k1 * y1 + k2 * y2 + k3 * y3 + k4 * y4

Per il calcolo del polinomio si scrive:

y = 1 / SQR(1 + (e ^ 2) * (S ^ 2))

da esse discende il programma:

```
LINE ( 0, 0 )-( 0, 320 )        ' ASSE Y  2 quadranti

LINE ( 0, 160 )-( 460, 160 ) ' ASSE X  2 quadranti

LOCATE 8, 66 : PRINT "e"  ' richiesta del parametro (e)

LOCATE 9, 66 : INPUT e   ' introduzione del parametro (e)

LOCATE 10, 66 : PRINT "n"   ' richiesta del valore di (n)

LOCATE 11, 66 : INPUT n  ' introduzione del valore di n

LOCATE 12, 66 : PRINT "x max"  ' richiesta del valore di x max

LOCATE 13, 66 : INPUT xm ' introduzione del valore di x max  ricerca del valore di n introdotto

IF  n = 1 THEN  k1 = 1  ELSE  k1 = 0  ' se n=1 pone k1=1 altrimenti pone k1=0

IF  n = 2  THEN  k2 = 1  ELSE  k2 = 0  ' se n=2 pone k2=1 altrimenti pone k2=0

IF  n = 3  THEN  k3 = 1  ELSE  k3 = 0  ' se n=3 pone k3=1 altrimenti pone k3=0

IF  n = 4  THEN  k4 = 1  ELSE  k4 = 0  ' se n=4 pone k4=1 altrimenti pone k4=0

FOR  x = 0  TO  xm  STEP  ( xm / 1000 )  ' gestisce il campo ed il passo della variabile indipendente  x

y1 = x                              ' calcolo delle quattro funzioni Cn(x)

y2 = ( 2 * ( x ^ 2 ) ) - 1          ' una sola delle quali

y3 = ( 4 * ( x ^ 3 ) - 3 * x )      ' sarà presentata

y4 = ( 8 * ( x ^ 4 ) - 8 * ( x ^ 2 ) + 1 )  ' sul video del P.C

S = k1 * y1 + k2 * y2 + k3 * y3 + k4 * y4  ' seleziona la funzione  Cn(x) in base al valore di n

y = 1 / SQR( 1 + ( e ^ 2 ) * ( S ^ 2 ) )  ' calcola il valore del polinomio in base alla Cn(x) selezionata

PSET ( 460 * x / xm , 160  -  160 * y ), 14  ' presenta il grafico del polinomio

NEXT x  ' rimanda all'istruzione FOR x= .... per il calcolo dei punti successivi
```

Per provare completamente il programma è necessario utilizzare tutte e quattro le funzioni Cn(x) per n = 1, n = 2, n = 3, n = 4 scegliendo di volta in volta un diverso valore del parametro (e) e dell'intervallo delle ascisse; iniziamo il lavoro per n = 1 e = 1 x max = 10:

```
F5
                                             e
                                             ? 1
                                             n
                                             ? 1
                                             x max
                                             ? 10
```

dopo l'introduzione dell'ultimo dato si ha la presentazione della prima curva caratteristica del polinomio di Chebyschev, il diagramma è riportato in figura 53.

Dalla figura si rileva che la curva non presenta alcuna ondulazione nella parte alta del tracciato e decresce molto dolcemente, ciò è dovuto al minimo valore dell'ordine del polinomio. Questo caso è utile soltanto come esempio didattico dato che la curva ottenuta non ha un effetto utile di mascheramento.

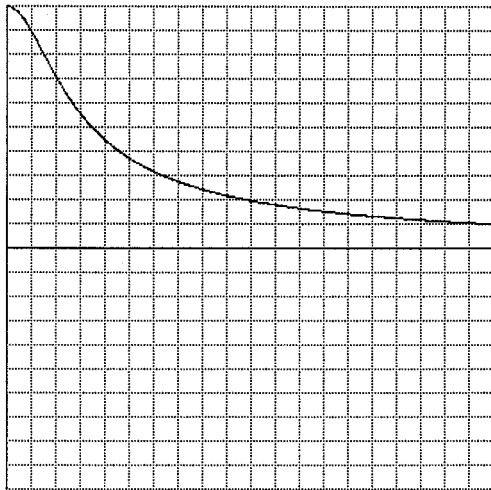



Figura 53
Andamento polinomio
di Chebyschev per e = 1, n = 1
Scala asse x = .5/div
Scala asse y = 1/div.

Risultati nettamente diversi e più importanti dei precedenti si ottengono con il polinomio di secondo ordine (n = 2) e con i valori: e = 1, x max = 10

F5

e
? 1
n
? 2
x max
? 10

Si ha la presentazione del polinomio come mostrato in figura 54, dalla curva si vede che è presente una ondulazione che si estende da y = .7 a y = 1 e che la pendenza è sensibile e tale da portare il tracciato al livello y = .13 per x = 2.
Questi risultati sono già utilizzabili a scopi di mascheramento.

Anche per il polinomio di Chebyschev, così come per il polinomio di Butterworth, la variabile indipendente deve essere espressa nel dominio della frequenza quando il polinomio, definendo le caratteristiche di un quadripolo, abbia funzione di mascheramento di fenomeni a carattere ondulatorio.

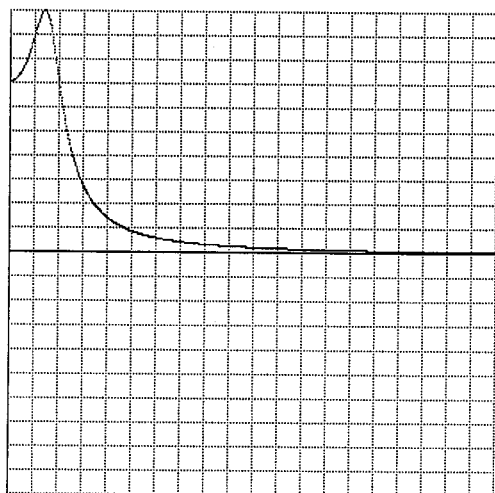



Figura 54
Andamento polinomio
di Chebyschev per e = 1, n =2
Scala asse x = .5/div.
Scala asse y = 1/div.

Un andamento del polinomio di Chebychev di notevole interesse applicativo si ha per n = 3 ed e = 5 con questi valori si ottengono ridotte ondulazioni nella parte alta del grafico e elevata pendenza, per evidenziare meglio il risultato conviene adottare un piccolo intervallo delle ascisse con x max = 5:

F5

e
? .5
n
? 3
x max
? 5

si ha la curva di figura 55 in cui l'ondulazione è contenuta tra y = .9 e y =1 e la pendenza del tratto discendente è tale che per x = 2 il valore di y si è attenuato a .08.
Un ultimo esempio si sviluppa per e = .2, n = 4, x max = 2

F5

e
? .2
n
? 4
x max
? 2

si ottiene l'ottimo andamento di mascheramento riportato in figura 56, il questo caso l'ondulazione

è ridotta a livelli appena percettibili e la pendenza del tratto discendente porta y =.05 per x = 2.

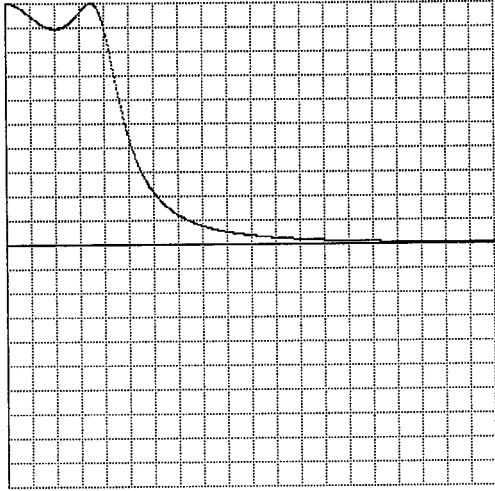



Figura 55
Andamento polinomio di Chebyschev per e =.5, n = 3
Scala asse x = .25/div.
Scala asse y = 1/div.

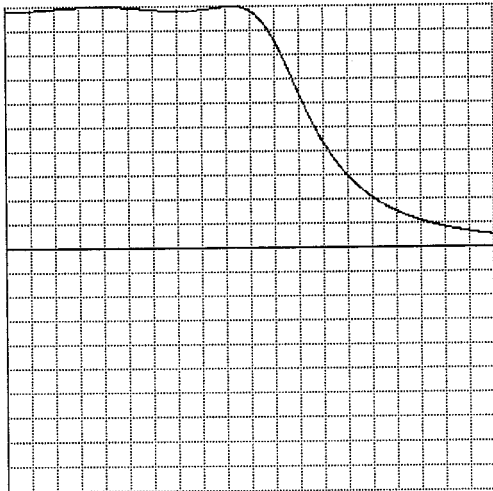



Figura 56
Andamento polinomio di Chebyschev per e =.2, n = 4
Scala asse x = .1/div.
Scala asse y = 1/div.

10. 6 La trasformazione semplice del polinomio di Chebychev

Il polinomio di Cebychev consente una trasformazione semplice che rovescia l'andamento della funzione rispetto all'originale.

La caratteristica della trasformazione del polinomio di Cebychev forma una sorta di gradino a salire da livelli molto piccoli di y, che iniziano per $x > 0$, per poi crescere rapidamente, da un certo valore di x in poi, giungendo in una zona a livello alto in cui sono presenti delle ondulazioni.
La trasformazione consiste nel cambiare la variabile x della funzione Cn(x) in xo/x, si ottiene così una nuova serie di funzioni:

n	Cn(x)
1	xo / x
2	$2 (xo / x)^2 - 1$
3	$4 (xo / x)^3 - 3 (xo / x)$
4	$8 (xo / x)^4 - 8 (xo / x)^2 + 1$
..	....

Dove il valore xo rappresenta l'ascissa a cui termina il tratto ascendente della curva per iniziare il tratto alto ondulato.
L'incidenza dei valori (e), (n), Cn(x), sull'andamento del polinomio trasformato è simile all'incidenza che questi hanno sul polinomio normale .
Nel compilare il nuovo programma di calcolo si deve evitare di inserire $x = 0$ nel campo di variabilità delle ascisse onde non creare condizioni di infinito.
Per la stesura del citato programma è necessario procedere alla composizione delle nuove Cn(x) secondo le note corrispondenze simboliche in Qbasic, seguendo i criteri di selezione delle funzioni Cn(x), al variare di n, già adottati nel paragrafo 10.5

n = 1; C1(x) = y1 = xo / x

n = 2; C2(x) = y2 = 2 * ((xo / x) ^ 2) - 1

n = 3; C3(x) = y3 = 4 * ((xo / x) ^ 3) - 3 * (xo / x)

n = 4; C4(x) = y4 = 8 * ((xo / x) ^ 4) - 8 * ((xo / x) ^ 2) + 1

Selezione di Cn(x)

-L'espressione di S è data da:

S = k1 * y1 + k2 * y2 + k3 * y3 + k4 * y4

Calcolo del polinomio

-L'espressione di y è:

y = 1 / SQR(1 + (e ^ 2) * (S ^ 2))

implementandole si ottiene il programma per il computo e la presentazione del polinomio in trasformazione semplice di Chebychev:

```
LINE ( 0, 0 )-( 0, 320 )        ' ASSE  Y   2  quadranti

LINE ( 0, 160 )-( 460, 160 )  ' ASSE  X  2  quadranti

LOCATE 8, 66 : PRINT "e"   ' richiesta del parametro (e)

LOCATE 9, 66 : INPUT  e   ' introduzione del parametro (e)

LOCATE 10, 66 : PRINT "n"    ' richiesta del valore di (n)

LOCATE 11, 66 : INPUT  n   ' introduzione del valore di n

LOCATE 12, 66 : PRINT "xo"    ' richiesta del valore di xo

LOCATE 13, 66 : INPUT  xo   ' introduzione del valore di xo

LOCATE 14, 66 : PRINT "x max"   ' richiesta del valore di x max

LOCATE 15, 66 : INPUT xm  ' introduzione del valore di x max

' ricerca del valore di n introdotto

IF  n = 1  THEN  k1 = 1  ELSE  k1 = 0  ' se n=1 pone k1=1 altrimenti pone k1=0

IF  n = 2  THEN  k2 = 1  ELSE  k2 = 0  ' se n=2 pone k2=1 altrimenti pone k2=0

IF  n = 3  THEN  k3 = 1  ELSE  k3 = 0  ' se n=3 pone k3=1 altrimenti pone k3=0

IF  n = 4  THEN  k4 = 1  ELSE  k4 = 0  ' se n=4 pone k4=1 altrimenti pone k4=0

FOR  x = .000001  TO  xm  STEP ( xm / 1000 )   ' gestisce il campo ed il passo della variabile
                                                ' indipendente di x con esclusione del valore  x=0

y1 =  xo / x                ' calcolo delle quattro funzioni  Cn(x)  trasformate

y2 = ( 2 * ( ( xo / x ) ^ 2 ) ) - 1                        ' una sola delle quali

y3 = ( 4 * ( ( xo / x ) ^ 3 ) - 3 * ( xo / x ) )           ' sarà presentata

y4 = ( 8 * ( ( xo / x ) ^ 4 ) - 8 * ( ( xo / x ) ^ 2 ) + 1 )  ' sul video del P.C

S = k1 * y1 +  k2 * y2 +  k3 * y3 + k4  * y4  ' seleziona la funzione  Cn(x) in base al valore di n

y = 1 / SQR( 1 + ( e ^ 2 ) * ( S ^ 2 ) )  ' calcola il valore del polinomio in base alla Cn(x) selezionata

PSET ( 460 * x / xm , 160 - 160 *  y), 14   ' presenta il grafico del polinomio   trasformato

NEXT x   ' rimanda all'istruzione FOR x= .... per il calcolo dei punti successivi
```

proviamo il nuovo programma per e = .5; n = 4; xo =10; x max = 20
premiamo F5

```
e
? .5
n
? 4
xo
? 10
x max
? 20
```

dopo l'introduzione dell'ultimo dato si ha la presentazione della curva caratteristica del polinomio di Chebychev trasformato, il diagramma è riportato in figura 57.
Dalla figura si rileva che la curva resta a livelli molto bassi per un certo tratto, per poi salire rapidamente fino a livello di y =.9 quando x = xo = 10, raggiunge successivamente il livello alto nel quale ondula modestamente. L'effetto di mascheramento in questo caso si realizza per valori inferiori a xo, cioè nell'intervallo tra x > 0 e x < xo, il polinomio trasformato ha di fatto un andamento rovesciato rispetto al polinomio originale.

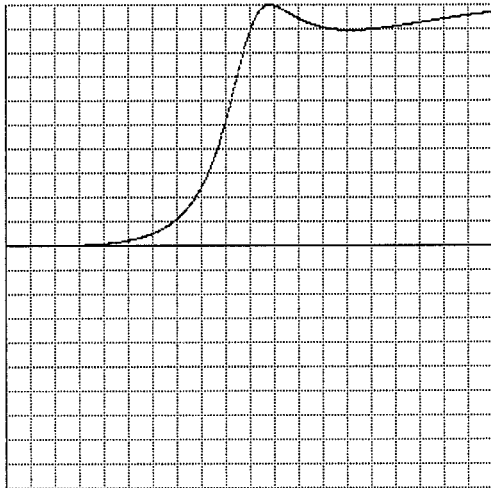



Figura 57
Andamento polinomio di Chebyschev trasformato simmetrico per e = .5, n = 4
Scala asse x = 1/div.
Scala asse y = 1/div.

10.7 La doppia trasformazione del polinomio di Chebychev

Anche il polinomio di Chebychev, così come quello di Butterworth, dà modo di ottenere una funzione a doppio scalino .
Questa caratteristica si ottiene dalla trasformazione delle funzioni Cn(x) in cui alla variabile x viene sostituita la variabile (x - xc) come sotto indicato:

n	Cn(x)
1	$x - xc$
2	$2 (x - xc)^2 - 1$
3	$4 (x - xc)^3 - 3 (x - xc)$
4	$8 (x - xc)^4 - 8 (x - xc)^2 + 1$

Dove il valore xc rappresenta l'ascissa corrispondente al punto centrale dello scalino, si può infatti posizionare lo scalino, ovvero le due zone di mascheramento laterali, dimensionando xc opportunamente.
L'incidenza dei valori (e), (n), Cn(x), sull'andamento del polinomio in doppia trasformazione è simile all'incidenza che questi hanno sul polinomio normale. Per la stesura del programma di calcolo e presentazione del grafico è necessario procedere alla composizione delle nuove Cn(x) secondo le corrispondenze simboliche in Qbasic, seguendo i criteri di selezione delle funzioni Cn(x), al variare di n, già adottati nel paragrafo 10.5:

Per le nuove Cn(x)

n = 1; C1(x) = y1 = x - xc

n = 2; C2(x) = y2 = 2 * ((x - xc) ^ 2) - 1

n = 3; C3(x) = y3 = 4 * ((x - xc) ^ 3) - 3 * (x - xc)

n = 4; C4(x) = y4 = 8 * ((x - xc) ^ 4) - 8 * ((x - xc) ^ 2) + 1

Per la selezione di Cn(x)

S = k1 * y1 + k2 * y2 + k3 * y3 + k4 * y4

Per il calcolo del polinomio

y = 1 / SQR(1 + (e ^ 2) * (S ^ 2))

implementando le espressioni sopra riportate si ottiene il programma per il computo e la presentazione del polinomio in doppia trasformazione di Cebychev:

```
LINE ( 0, 0 )-( 0, 320 )        ' ASSE Y 2 quadranti

LINE ( 0, 160 )-( 460, 160 )  ' ASSE X 2 quadranti

LOCATE 8, 66 : PRINT "e"  ' richiesta del parametro (e)

LOCATE 9, 66 : INPUT e    ' introduzione del parametro (e)

LOCATE 10, 66 : PRINT "n"   ' richiesta del valore di (n)

LOCATE 11, 66 : INPUT n    ' introduzione del valore di n

LOCATE 12, 66 : PRINT "xc"  ' richiesta del valore di xc

LOCATE 13, 66 : INPUT xc' introduzione del valore di xc

LOCATE 14, 66 : PRINT "x max"    ' richiesta del valore di x max

LOCATE 15, 66 : INPUT xm ' introduzione del valore di x max
                          ' ricerca del valore di n introdotto

IF n = 1 THEN k1 = 1 ELSE k1 = 0  ' se n=1 pone k1=1 altrimenti pone k1=0

IF n = 2 THEN k2 = 1 ELSE k2 = 0  ' se n=2 pone k2=1 altrimenti pone k2=0          &
```

```
IF n = 3  THEN  k3 = 1  ELSE  k3 = 0   'se n=3 pone k3=1 altrimenti pone k3=0

IF n = 4  THEN  k4 = 1  ELSE  k4 = 0   ' se n=4 pone k4=1 altrimenti pone k4=0

FOR x = .000001 TO xm STEP (xm / 1000)    ' gestisce il campo ed il passo della variabile indipendente di x

y1 =  x - xc                               ' calcolo delle quattro funzioni  Cn(x)  doppio trasformate

y2 = (2 * ( ( x - xc ) ^ 2) ) - 1          ' una sola delle quali

y3 = (4 * ( ( x - xc ) ^ 3) - 3 * ( x - xc ) )       ' sarà presentata

y4 = (8 * ( ( x - xc ) ^ 4) - 8 * ( ( x - xc ) ^ 2) + 1)   ' sul video del P.C

S = k1 * y1 + k2 * y2 + k3 * y3 + k4 * y4   ' seleziona la funzione  Cn(x) in base al valore di n

y = 1 / SQR( 1 + ( e ^ 2 ) * ( S ^ 2 ) )   ' calcola  polinomio in doppia trasformazione in base alla Cn(x) scelta

PSET ( 460 * x / xm , 160 - 160 * y ), 14 ' presenta il grafico del polinomio   trasformato

NEXT x   ' rimanda all'istruzione FOR x= .... per il calcolo dei punti successivi
```

Tracciamo il grafico del polinomio doppio trasformato per e =.5; n = 4; xc =10; x max = 20

```
F5        e
          ? .5
          n
          ? 4
          xc
          ? 10
          x max
          ? 20
```

dopo l'introduzione dell'ultimo dato si ha la presentazione della curva caratteristica del polinomio di Chebychev doppio trasformato, il diagramma è riportato in figura 58.

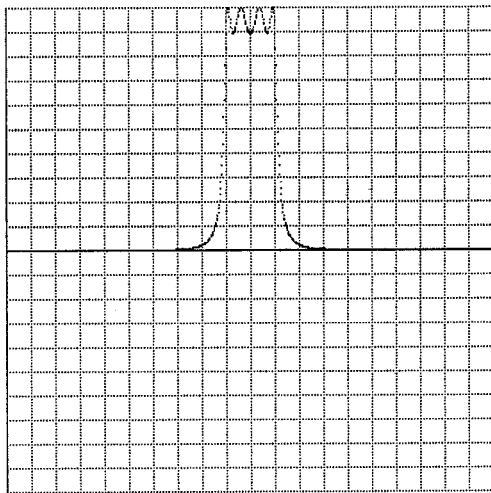



Figura 58
Andamento del polinomio di Chebychev doppio trasformato per e = .5, n = 4
Scala asse x = 1/div
Scala asse y = 1/div.

Dalla figura si osserva lo scalino piazzato al centro del reticolo, con il massimo dell'ondulazione del tratto a livello alto in corrispondenza di $x = xc = 10$. L'ondulazione è contenuta tra $y = .9$ e $y = 1$. I due tratti ad elevata pendenza, il primo a salire, il secondo a scendere, sono tali che per $xc-2$ e $xc+2$ il livello di y è già a valori molto piccoli. Questa curva mostra una ottima caratteristica di mascheramento per i due intervalli di x compresi, il primo, tra $x = 0$ e $x < xc-1$, il secondo per $x > xc+1$.

CAPITOLO 11
I NUMERI COMPLESSI

Il linguaggio Qbasic è di notevole aiuto nel calcolo e nella presentazione grafica dei numeri complessi. Questo capitolo è dedicato a tale argomento che normalmente non è trattato mediante P.C. Le soluzioni software che sono illustrate, anche se di semplice struttura, consentono di ottenere risultati molto interessanti in questo particolare settore della matematica.

11.1 La presentazione grafica di un numero complesso

La presentazione grafica di un numero complesso, di per sé cosa scontata, è la base per calcoli più complicati che senza l'ausilio del P.C. sarebbero gravosi da svolgere.
Iniziamo pertanto in questo paragrafo con esercitazioni riguardanti tale tipo di presentazione:
Dati ad esempio tre numeri complessi

$$N1 = 3 + j\,5$$
$$N2 = -7 + j\,4$$
$$N3 = 6 - j\,9$$

è possibile la loro rappresentazione grafica in un sistema di assi cartesiani a 4 quadranti mediante l'istruzione LINE (xo , yo) - (X , Y) in cui:

-nella prima parentesi sono indicate le coordinate del centro del reticolo, nel nostro caso xo = 230, yo = 160 (si veda il paragrafo 3.15).
-nella seconda parentesi sono indicate le variabili di posizione X e Y che, opportunamente rapportate, identificano rispettivamente la parte reale e la parte immaginaria dei numeri complessi.

Il segmento che l'istruzione traccia sullo schermo è il vettore del generico punto N che rappresenta un numero complesso, la lunghezza del segmento è il modulo del vettore .
Ricordando quanto spiegato nel paragrafo 3.15 in merito alla costruzione generale dell'istruzione PSET per gli assi cartesiani a 4 quadranti abbiamo che le coordinate X e Y dell'istruzione LINE sono date da:

$$X = 230 + k \cdot x \qquad Y = 160 - k2 \cdot y$$

dove x è la parte reale di N e y la parte immaginaria.
Per la presentazione dei numeri complessi dobbiamo determinare i valori di k e k2 in base rispettivamente alle loro parti reali e immaginarie; per mantenere la stessa scala per entrambe le parti e dovendo scegliere per i due assi il valore superiore della parte più elevata, è comodo per la suddivisione del reticolo assumere il valore 10, si ha perciò:

$k = 230 / 10 = 23 \qquad k2 = 160 / 10 = 16$

e le istruzioni per la presentazione dei tre punti assumono la struttura:

```
Per N1    LINE ( 230,160 ) - ( 230 + 23 * 3 , 160 - 16 * 5 )

Per N2    LINE ( 230,160 ) - ( 230 + 23 * ( -7 ) , 160 - 16 * 4 )

Per N3    LINE ( 230,160 ) - ( 230 + 23 * 6 , 160 - 16 * ( -9 ) )
```

Nel reticolo del sistema cartesiano è opportuno marcare l'asse delle ordinate con **+j** nel semipiano superiore e **-j** nel semipiano inferiore.

Possiamo compilare il semplice programma di presentazione di N come segue:

```
' FORMAZIONE DEL SISTEMA DI ASSI CARTESIANI A 4 QUADRANTI CON INDICAZIONI
' SULL'ASSE DELLE ORDINATE DEI SIMBOLI +j -j
SCREEN 9
FOR x = 0 TO 460 STEP 23
FOR y = 0 TO 320 STEP 2 ' VERTICALE
PSET (x, y), 7
NEXT y
NEXT x
LOCATE 1, 30 : PRINT "+j" ' presentazione del simbolo +j
LOCATE 23, 30: PRINT "-j" ' presentazione del simbolo -j
FOR y = 0 TO 320 STEP 16
FOR x = 0 TO 460 STEP 3 ' ORIZZONTALE
PSET (x, y), 7
NEXT x
NEXT y
LINE ( 230, 0 )-( 230, 320 ) ' ASSE Y 4 quadranti
LINE ( 0, 160 )-( 460, 160 ) ' ASSE X 4 quadranti
LINE ( 230,160 ) - ( 230 + 23 * 3 , 160 - 16 * 5 ),14 ' vettore colore giallo
LINE ( 230,160 ) - ( 230 + 23 * ( -7 ) , 160 - 16 * 4 ) ' vettore colore bianco
LINE ( 230,160 ) - ( 230 + 23 * 6 , 160 - 16 * ( -9 ) ), 3 ' vettore colore turchese
```

F5

è presentato in figura 59 il sistema di assi cartesiani a 4 quadranti in cui compaiono:
-nel primo quadrante il vettore giallo che supporta il punto N1 all'incrocio del reticolo a cui corrispondono le coordinate x = 3 e y = 5
-nel secondo quadrante il vettore bianco che supporta N2 all'incrocio del reticolo a cui corrispondono le coordinate x = -7 e y = 4
-nel quarto quadrante il vettore turchese che supporta N3 all'incrocio del reticolo a cui corrispondono le coordinate x = 6 e y = -9.

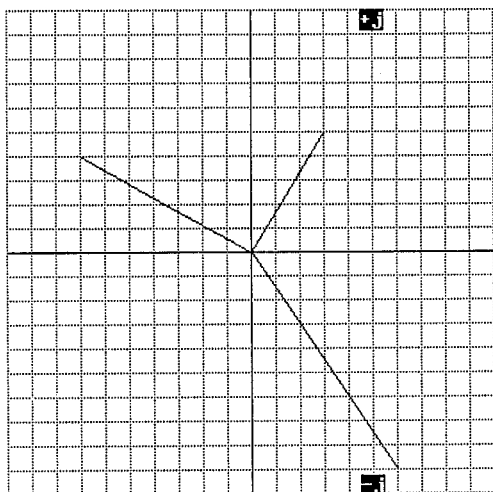



Figura 59
Presentazione grafica
di 3 numeri complessi

11.2 Le operazioni sui numeri complessi

Prima di entrare nel dettaglio di questo paragrafo ricordiamo che un numero complesso N = x + jy può essere espresso, sia in forma cartesiana, mediante i valori di x e y rispettivamente parte reale e parte immaginaria di N, sia in forma polare, mediante il modulo |M| e l'argomento Arg°= = Arcotang (y / x) di un **vettore** N. Pertanto, per completezza, i risultati delle operazioni forniranno sempre, sia la parte reale R e la parte immaginaria I, relativa alla rappresentazione cartesiana, sia il modulo |M| e l'argomento Arg° relativi alla rappresentazione polare.
Come è noto le quattro operazioni aritmetiche di somma, sottrazione, moltiplicazione e divisione sono estensibili anche ai numeri complessi, le prime due si possono eseguire facilmente con carta e matita, le altre invece richiedono un certo lavoro di calcolo. Il Qbasic ci aiuta in questo lavoro dando modo di sviluppare rapidamente tanto il calcolo delle diverse operazioni, quanto la presentazione grafica del numero complesso risultante dal calcolo stesso.
Vediamo come scrivere in linguaggio Qbasic le quattro formule di cui abbiamo parlato:
Dato che il Qbasic non riconosce i numeri complessi questi devono essere scomposti nelle parti:

Reale = R
Immaginaria = I

Se N = x + j y si ha:
R = x
I = y

soltanto in questo modo sono possibili le uguaglianze simboliche riportate nella pagina seguente.

Si deve osservare che l'uguaglianza simbolica relativa al calcolo dell'argomento, Arg°, del risultato di una operazione tra numeri complessi, è completata con l'addendo T che consente, al programma, di presentare i valori di Arg° sempre e soltanto come angoli positivi nel campo di variabilità compreso tra 0° e 360°.

operazione	simbologia ordinaria	simbologia Qbasic
somma	(x1+jy1) + (x2+jy2)	R = x1 + x2 ; I = y1 + y2
differenza	(x1+jy1) - (x2+jy2)	R = x1 - x2 ; I = y1 - y2
prodotto	(x1+jy1) · (x2+jy2)	R = x1 * x2 - y1 * y2 I = x1 * y2 + y1 * x2
quoziente	(x1+jy1) / (x2+jy2)	R = (x1 * x2 + y1 * y2) / (x2^2 + y2^2) I = (x2 * y1 - x1 * y2) / (x2^2 + y2^2)
modulo del risultato	$(R^2 + I^2)^{1/2}$	I M I = SQR(R ^ 2 +I ^ 2)
argomento del risultato	Arcotang (I / R)	Arg° = T + 57.2957 *ATN (I / R)

Come si evince dalle uguaglianze simboliche il Qbasic esegue le operazioni sempre e soltanto su R e su I, fornendo i risultati delle stesse nelle due parti separate. La presentazione dei vettori invece avviene come è stato mostrato del paragrafo 11.1.

11.3 Come implementare le quattro operazioni in un programma di calcolo e presentazione

La compilazione del programma di calcolo e presentazione delle quattro operazioni su due numeri complessi richiede una organizzazione delle istruzioni che è opportuno venga illustrata prima della stesura del programma stesso, essa si avvale di due nuove istruzioni **GOTO** ... ed **END**, l'istruzione (GOTO somma), ad esempio, che segue IF k = ..., invia direttamente l'esecuzione del programma alla routine di calcolo e presentazione denominata **somma:**, la routine di calcolo denominata **calcom:** si ferma, dopo i calcoli finali, incontrando l'istruzione di chiusura END

-Le prime 15 istruzioni sono dedicate alla formazione del sistema di assi cartesiani a 4 quadranti

-La 16ª e la 17ª istruzione sono dedicate all'ingresso del coefficiente di "scala", questo valore può essere aggiustato a piacere per far sì che i vettori dei punti siano presentati al meglio nell'ambito del reticolo

-Dalla18ª alla 21ª istruzione per l'introduzione dei valori della parte reale x1 e della parte immaginaria y1 relative al primo dei due numeri complessi

-Le istruzioni 22ª e 23ª permettono di stabilire quale tipo di operazione deve essere eseguita tra i due numeri complessi digitati in precedenza; con l'introduzione del valore voluto di k si potrà selezionare:

k = 1 per eseguire la somma

k = 2 per eseguire la differenza

k = 3 per eseguire il prodotto

k = 4 per eseguire il quoziente

-Dalla 24ª alla 27ª istruzione per l'introduzione dei valori della parte reale x2 e della parte immaginaria y2 relative al secondo dei due numeri complessi

-La 28ª istruzione per la presentazione grafica, in bianco, del vettore relativo al primo numero complesso, l'istruzione contiene i rapporti 230 / xm e 160 / xm che consentono di adattare al meglio la scala del reticolo alle dimensioni dei vettori

-La 29ª istruzione per la presentazione grafica, in bianco, del vettore relativo al secondo numero complesso, anche questa istruzione ha l'aggiustaggio della scala

-Dalla 30ª alla 33ª istruzione le strutture IF k = .. GOTO ... che impongono al programma di saltare direttamente, in base al valore digitato di k: 1; 2; 3; 4, rispettivamente alle routine di calcolo somma; diff; prod; div

-Dalla 34ª istruzione alla 38ª la routine di calcolo per la somma dei due numeri complessi con invio alla routine comune di calcolo e presentazione dei risultati

-Dalla 39ª istruzione alla 43ª la routine di calcolo per la differenza dei due numeri complessi con invio alla routine comune di calcolo e presentazione dei risultati

-Dalla 44ª istruzione alla 48ª la routine di calcolo per il prodotto dei due numeri complessi con invio alla routine comune di calcolo e presentazione dei risultati

-Dalla 49ª istruzione alla 53ª la routine di calcolo per la divisione dei due numeri complessi con invio alla routine comune di calcolo e presentazione dei risultati

-Dalla 54ª alla 69ª istruzione per la presentazione del vettore, giallo, delle parti reale R, immaginaria I, e del calcolo e presentazione del modulo |M| e dell'argomento Arg° del numero complesso risultante

Illustrato quanto sopra procediamo alla stesura del programma di calcolo e presentazione:

```
' FORMAZIONE DEL RETICOLO CON INDICAZIONI DI +j -j

SCREEN 9                    ' (1ª istruzione)

FOR x = 0 TO 460 STEP 23

FOR y = 0 TO 320 STEP 2  ' VERTICALE

PSET (x, y), 7

NEXT y

NEXT x

LOCATE 1, 30: PRINT "+j"

LOCATE 23, 30: PRINT "-j"

FOR y = 0 TO 320 STEP 16

FOR x = 0 TO 460 STEP 3  ' ORIZZONTALE                    &
```

```
PSET (x, y), 7

NEXT x

NEXT y

LINE ( 230, 0 )-( 230, 320 )  ' ASSE  Y  4  quadranti

LINE ( 0, 160 )-( 460, 160 )  ' ASSE  X  2 e 4 quadranti    (15a istruzione)

' INGRESSO DATI

LOCATE 1, 66 : PRINT "scala"  ' valore di scala   (16a istruzione)

LOCATE 2, 66 : INPUT xm       ' (17a istruzione)

LOCATE 3, 66 : PRINT "x1"     ' parte reale del primo numero complesso   (18a istruzione)

LOCATE 4, 66 : INPUT x1

LOCATE 5, 66 : PRINT "y1"     ' parte immaginaria primo numero complesso

LOCATE 6, 66 : INPUT y1       ' (21a istruzione)

LOCATE 7, 66 : PRINT "k"      ' selezione tipo operazione k=1 (somma) ;k=2 (differenza) ;

LOCATE 8, 66 : INPUT k        ' k=3 (prodotto); k=4 (divisione)  ( 22a  e  23a istruzione )

LOCATE 9, 66 : PRINT "x2"     ' parte reale del secondo numero complesso  (24a istruzione)

LOCATE 10, 66 : INPUT x2

LOCATE 11, 66 : PRINT "y2"  ' parte immaginaria del secondo numero complesso

LOCATE 12, 66 : INPUT y2    ' (27a istruzione)

' PRESENTAZIONE DEI DUE VETTORI RELATIVI AI DUE NUMERI COMPLESSI DIGITATI

LINE ( 230, 160 )-( 230 + ( 230 / xm ) * x1, 160 - (160 / xm ) * y1)  ' (28a istruzione)

LINE ( 230, 160 )-( 230 + ( 230 / xm ) * x2, 160 - (160 / xm ) * y2)  ' (29a istruzione)

' SELEZIONE DEL TIPO DI OPERAZIONE DA ESEGUIRE

IF k = 1 GOTO somma   ' (30a istruzione)

IF k = 2 GOTO diff

IF k = 3 GOTO prod

IF k = 4 GOTO div     ' (33a istruzione)

' ROUTINE DI CALCOLO  VETTORE SOMMA

somma:                ' (34a istruzione)

R = x1 + x2

I = y1 + y2                                                          &
```

```
LOCATE 13, 60 : PRINT "SOMMA"
GOTO calcom    ' (38ª istruzione)
' ROUTINE DI CALCOLO  VETTORE DIFFERENZA
diff:     ' (39ª istruzione)
R = x1 - x2
I = y1 - y2  '
LOCATE 13, 60 : PRINT "DIFFERENZA"
GOTO calcom    ' (43ª istruzione)
' ROUTINE DI CALCOLO  VETTORE PRODOTTO
prod:                 ' (44ª istruzione)
R = ( x1 * x2 - y1 * y2 )
I = ( x1 * y2 + y1 * x2 )
LOCATE 13, 60 : PRINT "PRODOTTO"
GOTO  calcom   ' (48ª istruzione)
' ROUTINE DI CALCOLO VETTORE QUOZIENTE
div:                ' (49ª istruzione)
R = ( x1 * x2 + y1 * y2 ) / ( ( x2 ) ^ 2 + ( y2 ) ^ 2 )
I = ( x2  * y1 - x1 * y2 ) /  ( ( x2 ) ^ 2 + ( y2 ) ^ 2 )
LOCATE 13, 60 : PRINT "QUOZIENTE"
GOTO calcom   ' (53ª istruzione)
' ROUTINE DI CALCOLO COMUNE E PRESENTAZIONE
calcom:        ' (54ª istruzione)
LINE( 230,160 )-( 230 + ( 230/xm ) * R , 160 - ( 160/xm ) * I ),14   ' presentazione vettore risultante
M = SQR ( R^2 + I^2 )  ' calcolo del modulo vettore risultante
IF (R = 0) AND (I < >0)  THEN R = .00000001#   ' se R=0 impone R  diverso da 0; evita blocco nel calcolo Arg
IF (R > 0) AND (I = 0)   THEN T = 0        ' sei istruzioni per individuare il quadrante nel quale
IF (R < 0) AND (I = 0)   THEN T = 180    ' calcolare Arg°
IF (I > 0) AND (R > 0)   THEN T = 0
IF (I > 0) AND (R < 0)   THEN T = 180
IF (I < 0) AND (R < 0)   THEN T = 180                                          &
```

```
IF (I < 0) AND (R > 0)  THEN T = 360

A = T + 57.2957 * ATN ( I / R)   ' calcolo Arg°

LOCATE 14, 60 : PRINT "R="; R

LOCATE 15, 60 : PRINT "I="; I

LOCATE 16, 60 : PRINT "|M|="; M

LOCATE 17, 60 : PRINT "Arg°="; A

END  ' (66ª istruzione)
```

Per comprendere bene l'impiego del programma che abbiamo sviluppato è necessario eseguire un esercizio per ciascun tipo di operazione ed ottenere vettori chiaramente visibili e leggibili nelle loro coordinate. Iniziamo pertanto ad eseguire la seguente somma tra numeri complessi anche se questo esercizio può essere fatto rapidamente a memoria:

$$(x1+jy1) + (x2+jy2) = (5 + j\,4) + (3 + j\,2)$$

dati i valori in gioco scegliamo un valore di scala 10, in questo modo ciascuna divisione del reticolo vale 1, dovendo eseguire una somma digiteremo per k il valore 1.

```
F5
                     scala
                     ? 10
                     x1
                     ? 5
                     y1
                     ? 4
                     k
                     ? 1
                     x2
                     ?3
                     y2
                     ? 2

                   SOMMA
                   R = 8
                   I = 6
                   |M| = 10
                   Arg° = 36.86985
```

Si ha la presentazione della parte reale R ed immaginaria I del risultato della somma, il modulo e l'argomento, nonché il tracciato dei due vettori addendi e del vettore somma così come mostrato nella figura 60.

Dalla figura si vedono chiaramente i vettori all'estremo dei quali sono collocati i punti che rappresentano i numeri complessi oggetto del nostro esercizio:

-il vettore bianco più esteso, relativo al primo numero complesso, ha l'estremo coincidente con il punto di coordinate 5; j4

-il vettore bianco meno esteso, relativo al secondo numero complesso, ha l'estremo coincidente con il punto di coordinate 3; j2

-il vettore giallo, relativo alla somma dei due numeri complessi, ha l'estremo coincidente con il punto di coordinate R = 8; I = j6
-l'ampiezza del modulo |M| del vettore somma è pari a 10 divisioni del reticolo

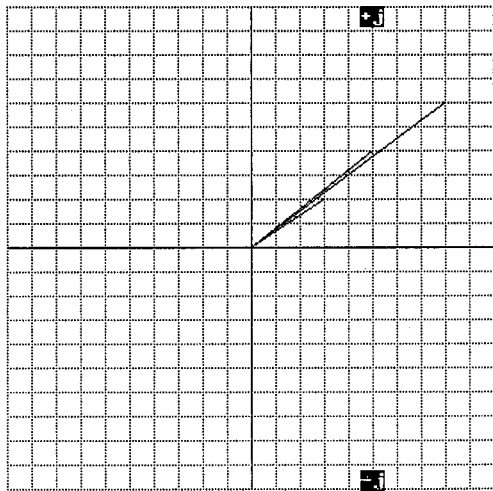



Figura 60
Rappresentazione grafica della somma tra due numeri complessi

Il secondo esercizio che andiamo a svolgere è la sottrazione tra numeri complessi:

$$(x1 + jy1) - (x2 + jy2) = (-9 + j\,7) - (1 - j\,2)$$

dati i valori in gioco scegliamo un valore di scala 10, in questo modo ciascuna divisione del reticolo vale 1, dovendo eseguire una differenza digiteremo per k il valore 2.

F5

```
scala
? 10
x1
? -9
y1
? 7
k
? 2
x2
?1
y2
? -2
DIFFERENZA
R = -10
I = 9
|M| = 13.45362
Arg = 138.0128
```

Si ha la presentazione della parte reale R ed immaginaria I del risultato della differenza, il modulo e l'argomento, nonché il tracciato dei due vettori, minuendo e sottraendo, e del vettore differenza così come mostrato nella figura 61 .
Dalla figura si vedono chiaramente i vettori all'estremo dei quali sono collocati i punti che rappresentano i numeri complessi oggetto del nostro esercizio:
-il vettore bianco più esteso, relativo al primo numero complesso, ha l'estremo coincidente con il punto di coordinate -9; j7
-il vettore bianco meno esteso, relativo al secondo numero complesso, ha l'estremo coincidente con il punto di coordinate 1; -j2
- il vettore giallo, relativo alla differenza dei due numeri complessi, ha l'estremo coincidente con il punto di coordinate R = -10; I = j9
- l'ampiezza del modulo |M| del vettore differenza è pari a circa 13.4 divisioni del reticolo

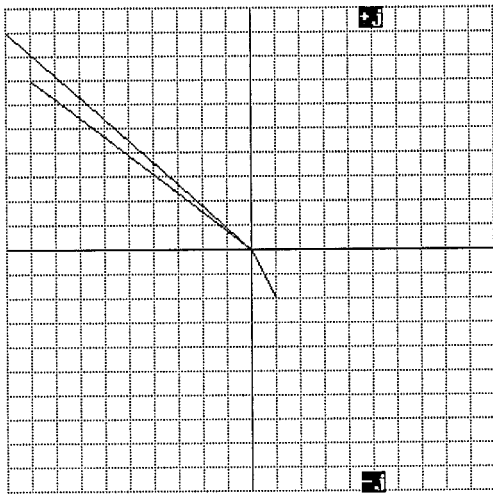



Figura 61
Rappresentazione grafica della differenza tra due numeri complessi

Se ripetiamo l'esercizio di sottrazione con due numeri complessi uguali si ha:

$$(x1 + jy1) - (x2 + jy2) = (1 + j2) - (1 + j2)$$

si ottiene, naturalmente, come risultato un vettore di modulo e argomento nulli.
In questo caso l'estremo del vettore risultante coincide con l'origine degli assi e la rappresentazione grafica non è significativa.

Il terzo esercizio che sviluppiamo è relativo al prodotto tra due numeri complessi:

$$(x1 + jy1) \cdot (x2 + jy2) = (1 + j 2) \cdot (3 + j 4)$$

dati i valori in gioco scegliamo un valore di scala 10, in questo modo ciascuna divisione del reticolo vale 1, dovendo eseguire un prodotto digiteremo per k il valore 3.

F5

scala
? 10
x1
? 1
y1
? 2
k
? 3
x2
?3
y2
? 4
PRODOTTO
R = - 5
I = 10
|M| = 11.18034
Arg° = 116.5651

Si ha la presentazione della parte reale R ed immaginaria I del risultato del prodotto il modulo e l'argomento, nonché il tracciato dei due vettori, fattori, e del vettore prodotto riportati in figura 62.
Dalla figura si vedono chiaramente i vettori all'estremo dei quali sono collocati i punti che rappresentano i numeri complessi oggetto del nostro esercizio:
-il vettore bianco meno esteso, relativo al primo numero complesso, ha l'estremo coincidente con il punto di coordinate 1; j2
-il vettore bianco più esteso, relativo al secondo numero complesso, ha l'estremo coincidente con il punto di coordinate 3; j4
-il vettore giallo, relativo al prodotto dei due numeri complessi, ha l'estremo coincidente con il punto di coordinate R= - 5; I = j10
-l'ampiezza del modulo |M| del vettore prodotto è pari a circa 11.2 divisioni del reticolo

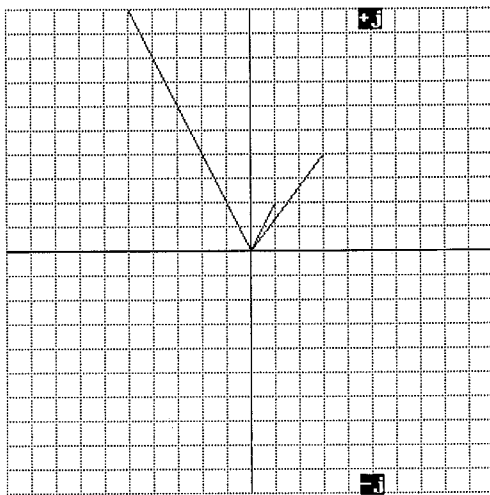



Figura 62
Rappresentazione grafica
del prodotto tra
due numeri complessi

Il quarto esercizio riguarda la divisione tra due numeri complessi:

$$(x1 + jy1) / (x2 + jy2) = (-7 + j\,22) / (2 + j\,3)$$

a prescindere dai valori in gioco scegliamo comunque un valore di scala 10 in favore della presentazione grafica del vettore quoziente, anche se ciò porterà fuori dal reticolo il primo numero complesso, in questo modo ciascuna divisione del reticolo vale 1, dovendo eseguire una divisione digiteremo per k il valore 4.

F5

scala
? 10
x1
? -7
y1
? 22
k
? 4
x2
?2
y2
? 3

QUOZIENTE
R = 4
I = 5
|M| = 6.403124
Arg° = 51.34012

Si ha la presentazione della parte reale R ed immaginaria I del risultato della divisione, il modulo e l'argomento, nonché il tracciato dei due vettori, dividendo e divisore, (il vettore dividendo esce fuori dal reticolo per avere la parte immaginaria superiore alla scala assegnata) e del vettore quoziente così come mostrato nella figura 63.
Dalla figura si vedono chiaramente i vettori all'estremo dei quali sono collocati i punti che rappresentano i numeri complessi oggetto del nostro esercizio:
-il vettore bianco più esteso, relativo al primo numero complesso, ha l'estremo che esce dal reticolo dato che ha coordinate -7; j 22
-il vettore bianco meno esteso, relativo al secondo numero complesso, ha l'estremo coincidente con il punto di coordinate 2; j 3
-il vettore giallo, relativo alla divisione dei due numeri complessi, ha l'estremo coincidente con il punto di coordinate R = 4; I = j 5
-l'ampiezza del modulo |M| del vettore quoziente è pari a circa 6.4 divisioni del reticolo

11.4 Osservazioni in merito al valore di scala

Nell'ultimo esercizio del paragrafo 11.3 abbiamo visto che la scelta del valore di scala ha impedito la presentazione completa di uno dei vettori coinvolti nell'operazione di divisione. La situazione che si è verificata è stata deliberatamente voluta, sia per consentire una operazione con risultati R ed I a valori interi, sia per mostrare che a volte alcuni vettori possono uscire dal reticolo se il valore di scala non è per loro dimensionato. E' chiaro che adottando valori di scala molto più elevati dei termini che costituiscono i numeri complessi in gioco nessuno di questi uscirà dal reticolo ma si avranno magari le tracce dei vettori troppo piccole e poco visibili . Il problema può essere risolto

per il vettore risultante, facendo girare il programma almeno due volte, la prima volta con valore di scala elevato si verificano i risultati numerici in R; I. Visti questi valori si ripete la routine di programma con un valore di scala che permetta la migliore presentazione del vettore risultante dall'operazione .

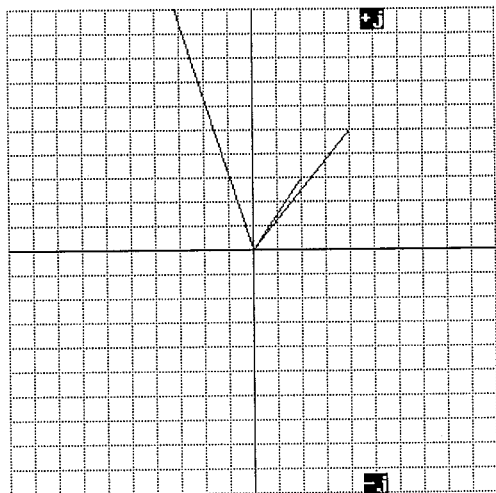



Figura 63
Rappresentazione grafica del quoziente tra due numeri complessi

11.5 Precisazioni sulla lettura degli argomenti

Il programma che abbiamo implementato sviluppa il calcolo, tra gli altri elementi, dell'argomento del numero complesso ottenuto come risultato delle operazioni, questo valore espresso in gradi sessagesimali mediante il simbolo Arg° è stato riportato automaticamente alla fine di ciascun esercizio del paragrafo 11.3. Il valore di Arg° che misura, in senso antiorario, l'angolo formato dal vettore con l'asse dei numeri reali positivi, non è facilmente valutabile sul tracciato dato che sarebbe necessaria una misura goniometrica sullo schermo del P.C.
Per rendere concreta, almeno una volta, la corrispondenza tra il valore dell'argomento scaturito dal calcolo e il valore dell'argomento visibile sul tracciato è utile un semplice esercizio di differenza tra due numeri complessi che ha come risultato un numero complesso dotato di argomento Arg° = 45, in questo caso vedremo che il vettore risultante biseca esattamente il primo quadrante denunciando un argomento pari a (90 / 2) °.

Con l'ausilio del programma del paragrafo 11.3 eseguiamo l'operazione:

$$(x1 + jy1) - (x2 + jy2) = (6 + j\,5) - (2 + j\,1)$$

F5

scala
? 10
x1
? 6
y 1

?5
k
?2
x2
?2
y2
?1
DIFFERENZA
R = 4
I = 4
|M| = 5.656854
Arg°= 44.99994

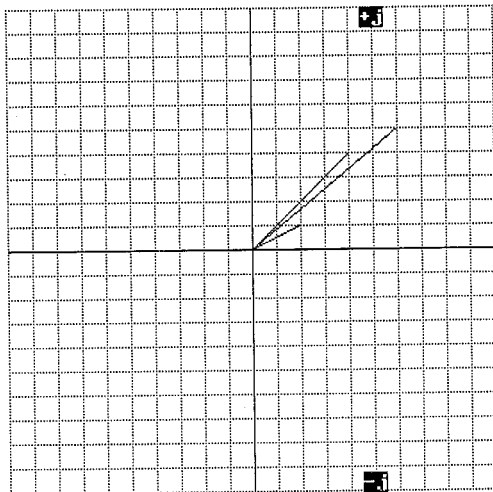



Figura 64
Rappresentazione grafica
della differenza tra
due numeri complessi
Vettore risultante Arg° = 45

Otteniamo il grafico di figura 64 dove si vedono chiaramente i vettori all'estremo dei quali sono collocati i punti che rappresentano i numeri complessi oggetto del nostro esercizio, in particolare il vettore giallo, relativo al risultato della sottrazione dei due numeri complessi, ha l'estremo coincidente con il punto di coordinate R = 4; I = j 4 e taglia in due il primo quadrante formando un angolo di 45° così come espresso numericamente dal calcolo di Arg° mediante la presentazione del valore Arg°= 44.99994.

11.6 I numeri complessi funzione di un'unica variabile reale

Sovente si incontrano numeri complessi, le cui parti reale, ed immaginaria, sono in funzione di una variabile indipendente che ne condiziona il posizionamento nel sistema di assi cartesiani.
Infatti se un numero complesso
N = x1 + j y1
è legato ad s tramite le funzioni
x1 = f1(s); y1 = f2(s)
ad ogni valore di s corrisponderà un numero complesso e, a questo, un punto rappresentativo come estremo del vettore che lo caratterizza.
Le due parti di N possono ad esempio dipendere da s secondo le funzioni:

$x1 = (s^3 \ / \ 30 \ ) + .1$

$y1 = (\text{Sen } s)^3 \ + .2$

ci saranno pertanto, per ogni valore assegnato alla variabile indipendente s, una coppia di valori x1 ed y1 che individueranno un ben determinato numero complesso x1+j y1; per s = 1.5 radianti, ad esempio, si ha

$x1 = (1.5^3 \ / \ 30) + .1 = .2125$

$y1 = (\text{Sen } 1.5)^3 \ + \ .2 = 1.1925$

ed N assume il valore: N = .2125 + j 1.1925

Se ora vogliamo rappresentare graficamente una serie di 10 numeri complessi generati dalle funzioni sopra espresse per la s variabile da -3 radianti a + 3 radianti dobbiamo ricorrere al programma illustrato nel paragrafo 11.1 apportando ad esso alcune modifiche, sia per calcolare per ciascun valore di s la corrispondente coppia x1;y1, sia per evitare di dover digitare 10 istruzioni del tipo LINE per il tracciamento dei 10 vettori calcolati. Per queste modifiche dobbiamo anzitutto trasformare in Qbasic le funzioni date:

simbologia ordinaria	**simbologia Qbasic**
$x1 = (s^3 / 30) + .1$	x1 = ((s ^ 3) / 30) + .1
$y1 = (\text{Sen } s)^3 + .2$	y1 = ((SIN (s)) ^ 3) + .2

e di seguito compilare il programma commentato con le modifiche e gli adattamenti di scala necessari:

```
' FORMAZIONE DEL SISTEMA DI ASSI CARTESIANI A 4 QUADRANTI CON INDICAZIONI

' SULL'ASSE DELLE ORDINATE DEI SIMBOLI +j ; -j

SCREEN 9

FOR x = 0 TO 460 STEP 23

FOR y = 0 TO 320 STEP 2  ' VERTICALE

PSET (x, y), 7

NEXT y

NEXT x                                                      &
```

```
LOCATE 1, 30: PRINT "+j"    ' presentazione del simbolo +j

LOCATE 23, 30: PRINT "-j"   ' presentazione del simbolo -j

FOR y = 0 TO 320 STEP 16

FOR x = 0 TO 460 STEP 3  ' ORIZZONTALE

PSET (x, y), 7

NEXT x

NEXT y

LINE ( 230, 0 )-( 230, 320 )  ' ASSE Y  4 quadranti

LINE ( 0, 160 )-( 460, 160 )  ' ASSE X  4 quadranti

' CALCOLO DEI VALORI DI x1 e y1  IN FUNZIONE DI s

FOR s = -3 TO 3 STEP .6 ' stabilisce il campo e l'incremento della variabile indipendente s

x1 = ( ( s ^ 3 ) / 30 ) + .1  ' calcola la parte reale del numero complesso funzione di s

y1 = ( ( SIN ( s ) ) ^ 3 ) + .2 ' calcola la parte immaginaria del numero complesso funzione di s

' COMPOSIZIONE ISTRUZIONE LINE SULLA BASE DEI VALORI CALCOLATI DI x1 E y1

LINE ( 230,160 ) - ( 230 + 230 * x1 , 160 - 130 * y1),14   ' traccia i vettori calcolati in giallo

NEXT s   ' rimanda all'istruzione FOR s=... per il calcolo dei successivi numeri complessi
```

Come si può vedere il programma è molto semplice, non resta che provarlo: F5
Viene presentato il grafico di figura 65 in cui compaiono, sul reticolo, un ventaglio di 10 vettori.

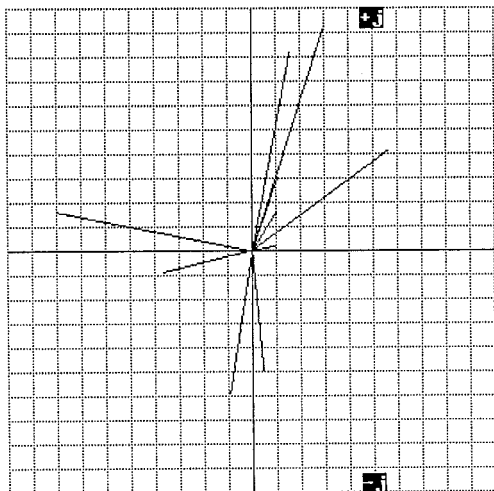



Figura 65
Rappresentazione grafica
di numeri complessi funzione
di variabile reale.
metodo dei vettori

La presentazione grafica dei numeri complessi che abbiamo ora mostrato è di importanza fondamentale nello studio di alcuni fenomeni fisici, essa permette di inquadrare l'andamento complessivo dei vettori, funzioni della variabile indipendente s, sia nella sequenza di rotazione dell'argomento, sia nella variazione del modulo. Questo tipo di grafica non si adatta bene nel caso in cui i vettori ruotino tanto da intersecare altri vettori, in tal caso si deve ricorrere ad una soluzione diversa.
La soluzione consiste nell'adottare l'istruzione PSET invece dell'istruzione LINE, rinunciando alla grafica dei vettori in favore della grafica dei punti nel piano complesso.
Se ripetiamo l'esempio precedente con l'istruzione PSET abbiamo il seguente programma modificato che pone in evidenza soltanto gli estremi dei vettori collocati nell'ambito del sistema di assi cartesiani a 4 quadranti.

```
SCREEN 9

FOR x = 0 TO 460 STEP 23

FOR y = 0 TO 320 STEP 2  ' VERTICALE

PSET (x, y), 7

NEXT y

NEXT x

LOCATE 1, 30: PRINT "+j"   ' presentazione del simbolo +j

LOCATE 23, 30: PRINT "-j"   ' presentazione del simbolo -j

FOR y = 0 TO 320 STEP 16

FOR x = 0 TO 460 STEP 3   ' ORIZZONTALE

PSET (x, y), 7

NEXT x

NEXT y

LINE ( 230, 0 )-( 230, 320 )   ' ASSE Y 4 quadranti

LINE ( 0, 160 )-( 460, 160 )   ' ASSE X 4 quadranti

' CALCOLO DEI VALORI DI x1 e y1 IN FUNZIONE DI s

FOR s = -3 TO 3 STEP .6 ' stabilisce il campo e l'incremento della variabile indipendente s

x1 = ( ( s ^ 3 ) / 30 ) + .1 ' calcola la parte reale del numero complesso funzione di s

y1 = ( ( SIN ( s ) ) ^ 3 ) + .2  ' calcola la parte immaginaria del numero complesso funzione di s

' COMPOSIZIONE ISTRUZIONE PSET SULLA BASE DEI VALORI CALCOLATI DI x1 E y1

PSET ( 230 + 230 * x1 , 160 - 130 * y1 ),14   ' insieme di 5 istruzioni per tracciare i punti

PSET ( 230 + 230 * x1 + 1 , 160 - 130 * y1 ),14  ' calcolati mediante crocette

PSET ( 230 + 230 * x1 - 1 , 160 - 130  * y1),14                                   &
```

```
PSET ( 230 + 230 * x1 , 160 - 130 * y1 + 1 ),14

PSET ( 230 + 230 * x1 , 160 - 130 * y1 - 1 ),14

NEXT s   ' rimanda all'istruzione FOR s=... per il calcolo dei successivi numeri complessi
```

F5

A seguito dell'azione su F5 il programma rende visibili i 10 punti oggetto del calcolo.

La nuova grafica di presentazione è mostrata in figura 66.

Dalla figura si vede che compaiono soltanto una sequenza di punti che descrivono le posizioni dei 10 numeri nel piano complesso, questi sono gli estremi dei vettori i cui moduli ora non sono visualizzati. Vedremo in seguito come questo tipo di grafica sia indispensabile per visualizzare particolari serie di numeri complessi. E' superfluo sottolineare che i punti che compaiono nel piano complesso **non sono** funzioni dei valori dell'asse delle ascisse, come nel caso delle funzioni di tabella, ma sono legati soltanto dalla variabile indipendente s che non compare nel grafico in alcun modo.

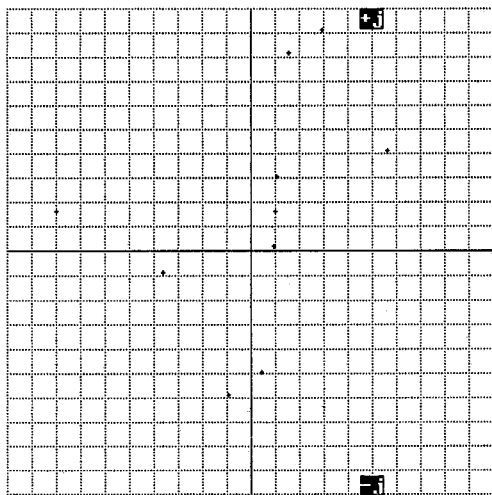



Figura 66
Rappresentazione grafica di numeri complessi funzione di variabile reale metodo dei punti

11.7 La grafica dei numeri complessi di un sistema risonante

L'impiego della grafica Qbasic ha una delle sue più interessanti applicazioni nella visualizzazione dei numeri complessi che esprimono l'andamento di un fenomeno fisico relativo ai sistemi risonanti in generale. Del fenomeno in questione mostriamo soltanto l'algoritmo complesso che ne rappresenta l'ammettenza, senza entrare nel merito delle problematiche teoriche e concettuali che lo riguardano.Il numero complesso in oggetto può essere scritto nel modo solito:

$$N = x1 + j\,y1$$

mentre la dipendenza delle parti reale ed immaginaria dalla variabile indipendente s è data da:

$$x1 = \frac{1}{1 + (s - 1/s)^2} + 1$$

$$y1 = s - \frac{s - 1/s}{1 + (s - 1/s)^2}$$

Volendo eseguire come esercizio la rappresentazione grafica di questo numero complesso, al variare di s, le due funzioni devono essere trasformate in Qbasic per l'implementazione nel programma, si ha perciò:

x1 = (1 / (1 + (s -1/s) ^ 2)) + 1

y1 = s - ((s - 1/s) / (1 + (s - 1/s) ^ 2))

La compilazione del programma deve essere preceduta da alcune definizioni numeriche:

Per rendere perspicuo l'esercizio si sceglie un campo di variabilità della s compreso tra s =.1 e s = 2.5 con passi di .1 in modo da calcolare 25 numeri complessi.
Si fissa per l'istruzione PSET: k = 230 / 4 = 57.5 e k2 = 160 / 4 = 40

Con questi valori e con le funzioni trasformate in Qbasic si compila il programma:

```
SCREEN 9

FOR x = 0 TO 460 STEP 23

FOR y = 0 TO 320 STEP 2  ' VERTICALE

PSET (x, y), 7

NEXT y

NEXT x

LOCATE 1, 30: PRINT "+j"    ' presentazione del simbolo +j

LOCATE 23, 30: PRINT "-j"   ' presentazione del simbolo -j

FOR y = 0 TO 320 STEP 16

FOR x = 0 TO 460 STEP 3  ' ORIZZONTALE

PSET (x, y), 7

NEXT x                                                          &
```

```
NEXT y
LINE ( 230, 0 )-( 230, 320 )  ' ASSE Y  4  quadranti
LINE ( 0, 160 )-( 460, 160 )  ' ASSE X  4  quadranti
' CALCOLO DEI VALORI DI x1 e y1  IN  FUNZIONE  DI s
FOR  s = .1  TO  2.5  STEP  .1   ' stabilisce il campo e l'incremento della variabile indipendente s
x1 = ( 1 / ( 1 + (  s -1/s ) ^ 2  ) ) + 1   ' calcolo della parte reale funzione di s
y1 = s - ( ( s - 1/s ) / ( 1 + ( s - 1/s ) ^ 2 ) )   ' calcolo della parte immaginaria funzione di s
' COMPOSIZIONE ISTRUZIONE  PSET  SULLA BASE DEI VALORI CALCOLATI DI x1 E y1
PSET  ( 230 + 57.5 * x1 , 160 -  40 * y1 ),14          ' insieme di 5 istruzioni
PSET  ( 230 + 57.5 * x1 + 1 , 160 -  40  * y1 ),14   ' per tracciare i punti calcolata mediante
PSET  ( 230 + 57.5 * x1 - 1 , 160 -  40 * y1 ),14     ' crocette
PSET  ( 230 + 57.5 * x1 , 160 -  40 * y1 + 1 ),14
PSET  ( 230 + 57.5 * x1 , 160 -  40 * y1 - 1 ),14
NEXT s    ' rimanda all'istruzione FOR s=... per il calcolo dei successivi numeri complessi

F5
```

L'esecuzione del programma porta al grafico di figura 67.

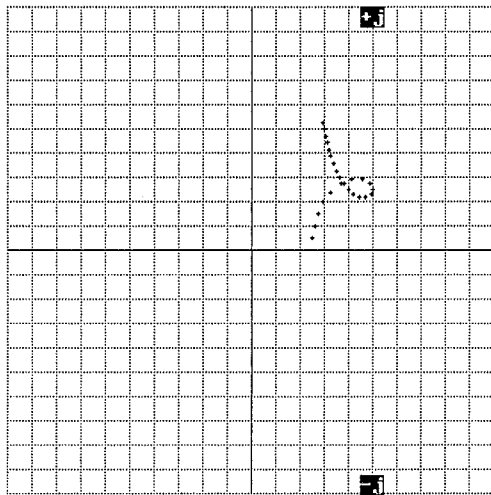



Figura 67
Rappresentazione grafica di numeri complessi funzione di variabile reale ammettenza di un sistema risonante

Il grafico ottenuto mostra la notevole capacità espressiva del mezzo di presentazione; si osserva, tracciato nel primo quadrante, l'insieme dei 25 punti disposti secondo un pseudo nodo che mette in evidenza, all'occhio esperto, la caratteristica di risonanza del sistema fisico.
Da questo esercizio si comprende bene che se avessimo tracciato i 25 vettori questi avrebbero completamente coperto il nodo cancellando di fatto le informazioni che invece sono state fornite.
Si conclude che il sistema grafico deve essere adattato, con un poco di esperienza, al tipo di informazioni che si desiderano dal calcolo.
Si deve inoltre osservare che aver limitato a 25 i punti sul piano consente di associare, a vista, a ciascuno di essi il valore della variabile indipendente che lo ha generato; infatti il primo punto verso l'asse delle ascisse rappresenta il numero complesso ottenuto per s = .1, il punto successivo, verso l'alto, rappresenta il numero complesso ottenuto per s =.2, e così via fino al punto più in alto generato dal valore di s = 2.5. La possibilità di questa importantissima correlazione visiva, tra valore della variabile indipendente s e posizione dei punti nel piano complesso, è una ulteriore fonte di informazione che non si potrebbe avere se il numero dei punti fosse talmente elevato da fornire una curva continua invece che una netta punteggiata.
Si può pertanto concludere che il sistema grafico, se correttamente utilizzato, è in grado di fornire con immediatezza una quantità di informazioni riguardanti il sistema risonante, che non sarebbero facilmente acquisibili tramite un tabulato numerico che raccogliesse i 25 numeri complessi, sia che fossero espressi in forma cartesiana, sia che fossero espressi in forma polare.

11.8 La grafica complementare dei numeri complessi di un sistema risonante

Per completare l'esposizione delle metodologie grafiche e numeriche relative all'impiego dei numeri complessi per lo studio di un sistema risonante è necessario accennare alla grafica complementare che rappresenta un utile corollario all'esposizione fatta nel paragrafo 11.7.
Per ricavare ulteriori informazioni dalla serie dei numeri complessi che rappresentano un fenomeno fisico sono disponibili altre versioni grafiche delle stesse per evidenziarne:

-andamento della parte reale in funzione di s
-andamento della parte immaginaria in funzione di s
-andamento del modulo in funzione di s
-andamento dell'argomento in funzione di s

Queste funzioni di s, che caratterizzano la serie dei numeri complessi, sono funzioni nel campo reale e possono essere rappresentate con un numero elevato di punti dato che in questo caso la variabile indipendente si trova collocata sulle ascisse e il riferimento con i punti tracciati è immediato.
Per completare l'esercizio del paragrafo 11.7 impostiamo il nuovo programma per il tracciamento della parte reale x1 come segue:
Dal grafico di figura 67 si rileva che il valore più elevato della parte reale si ha all'estrema destra del nodo per s = 1 a cui corrisponde x1 = 2, questo è il valore sul quale deve essere adattata la scala delle ordinate .
Per la scala delle ascisse il campo di s dovrà avere gli stessi valori imposti nell'esercizio sopra citato: da s =.1 a s = 2.5.
La figura 67 mostra inoltre che il campo di variabilità di x1 è sempre positivo, si adatta perciò una presentazione ad un solo quadrante, per la quale l'istruzione PSET diventa:

PSET(k1 * s , 320 - k3 * x1)   dove

k1 = 460 / s = 460 / 2.5 = 184
k3 = 320 / x1 = 320 / 2 = 160

Ciò premesso abbiamo:

```
LINE ( 0 , 320 ) - ( 460 , 320 )  ' comanda il tracciamento dell'asse delle ascisse (asse  s)
                                  ' per coordinate ad 1 quadrante

LINE ( 0 , 0 ) - ( 0 , 320 )      ' comanda il tracciamento dell'asse delle ordinate ( asse  x1)
                                  ' per coordinate ad 1  quadrante

LOCATE 1,66 : PRINT " PARTE REALE"    ' indicazione della funzione presentata

FOR  s = .1  TO  2.5  STEP .001  '  campo di variabilità ed incremento della s per avere 2500 punti di calcolo

x1 = ( 1 / ( 1 + (  s -1/s ) ^ 2  ) ) + 1  ' calcolo della parte reale funzione di s

PSET( 184  * s ,  320 - 160 * x1 ),14   ' presentazione della parte reale come una curva continua

NEXT s ' comanda il programma al ritorno automatico  all'istruzione FOR s =  ecc.

                F5
```

si ha la presentazione del grafico dell'andamento della parte reale in funzione di s così come mostrato in figura 68.
La figura evidenzia molto chiaramente come varia la parte reale del numero complesso al variare di s.
La funzione cresce, con l'incremento di s, ed ha il suo massimo per s = 1 dopo di che decresce con regolarità .
In questo caso la curva è continua e il legame tra x1 ed s è diretto; si può pertanto leggere, con l'ausilio del reticolo, che valore assume la parte reale del numero complesso per un qualsiasi valore di s compreso nel campo di variabilità assegnato.
Informazioni di questo tipo, con la precisione offerta dalla nuova grafica, non sono certamente ricavabili dalla figura 67 che ha invece lo scopo preciso di fornire un andamento globale della serie dei numeri complessi.

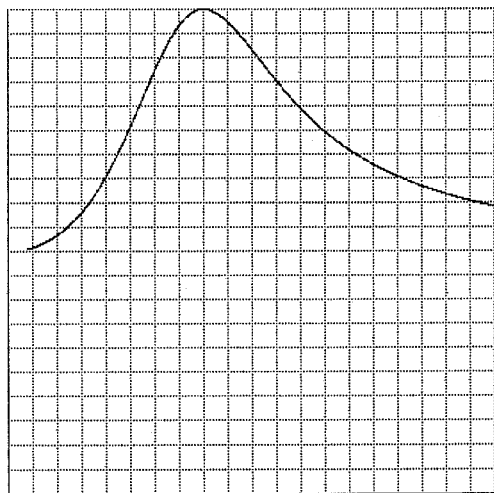



Figura 68
Rappresentazione grafica
della parte reale
dell'ammettenza
del sistema risonante

Analogamente alla parte reale si traccia l'andamento della parte immaginaria relativa alla serie dei numeri complessi sulla base dei seguenti dati:
Dal grafico di figura 67 si rileva che il valore più elevato della parte immaginaria si ha all'estremo superiore della punteggiata per s = 2.5 a cui corrisponde y1 = 2.1 che per facilitare la lettura del grafico si porta a 2.5, questo è il valore sul quale deve essere adattata la scala delle ordinate.
Per la scala delle ascisse il campo di s dovrà avere gli stessi valori imposti in precedenza: da s =.1 a s = 2.5.
La figura 67 mostra inoltre che il campo di variabilità di y1 è sempre positivo, si adatta perciò una presentazione ad un solo quadrante, per la quale l'istruzione PSET diventa
PSET(k1 * s , 320 - k3 * x1) dove k1 = 460 / s = 460 / 2.5 = 184
k3 = 320 / y1 = 320 / 2.5 = 128

Il programma per la presentazione della parte immaginaria sarà pertanto:

```
LINE ( 0 , 320 ) - ( 460 , 320 )  ' comanda il tracciamento dell'asse delle ascisse (asse s)
                                  ' per coordinate ad 1 quadrante

LINE ( 0 , 0 ) - ( 0 , 320 )  ' comanda il tracciamento dell'asse delle ordinate ( asse y1)
                              ' per coordinate ad 1 quadrante

LOCATE 1,66: PRINT "PARTE IMMAG."  ' indicazione della funzione presentata

FOR s = .1 TO 2.5 STEP .001  ' campo di variabilità ed incremento della s per avere 2500 punti di calcolo

y1 = s - ( ( s - 1/s ) / ( 1 + ( s - 1/s ) ^ 2 ) )  ' calcolo della parte immaginaria funzione di s

PSET( 184 * s , 320 - 128 * y1 ),14  ' presentazione della parte immaginaria come una curva continua

NEXT s ' comanda il programma al ritorno automatico all'istruzione FOR s = ecc.
```

Premendo F5 si ha la presentazione del grafico dell'andamento della parte immaginaria in funzione di s così come mostrato in figura 69.

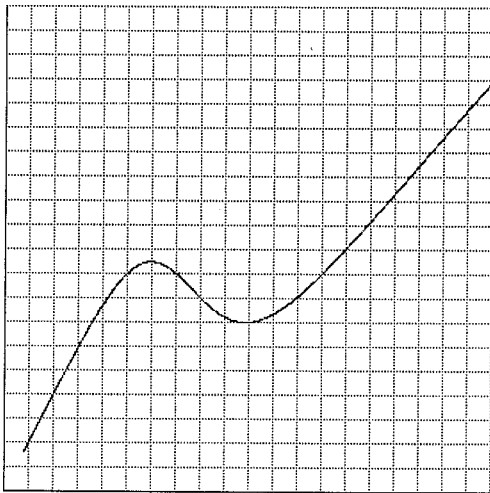



Figura 69
Rappresentazione grafica della parte immaginaria dell'ammettenza del sistema risonante

La figura evidenzia come varia la parte immaginaria del numero complesso al variare di s.
Anche in questo caso la curva è continua e il legame tra y1 ed s è diretto, si può pertanto leggere, con l'ausilio del reticolo, che valore assume la parte immaginaria del numero complesso per un qualsiasi valore di s compreso nel campo di variabilità assegnato. La funzione ha un punto di massimo per s = .75 ed un punto di minimo per s = 1.2.
Per la presentazione grafica del modulo della serie di numeri complessi si deve iniziare, come per le altre parti, dall'esame del grafico globale di figura 67 in cui risulta che la distanza massima tra l'origine degli assi e la punteggiata si ha per il punto corrispondente al valore di s = 2.5; sulla base di questo numero si deve calcolare il massimo valore del modulo con il programma:

```
CLS

INPUT " s= " ; s

x1 = ( 1 / ( 1 + ( s - 1/s ) ^ 2 ) ) + 1

y1 = s - ( ( s - 1/s ) / ( 1 + ( s - 1/s ) ^ 2 ) )

M = SQR( x1 ^ 2 + y1 ^ 2 )

PRINT "|M|=" ; M
```

che per s = 2.5 dà |M| = 2.4 che si arrotonda per comodità di lettura delle scale in 2.5.
Per la scala delle ascisse il campo di s dovrà avere gli stessi valori imposti in precedenza:
da s =.1 a s = 2.5.
Dato che il modulo di un numero complesso è sempre positivo, si adotta anche in questo caso una presentazione ad un solo quadrante, per la quale l'istruzione PSET diventa

PSET(k1 * s , 320 - k3 * x1) dove k1 = 460 / s = 460 / 2.5 = 184
k3 = 320 / y1 = 320 / 2.5 = 128

Il programma per la presentazione dell'andamento del modulo è:

```
LINE ( 0 , 320 ) - ( 460 , 320 )  ' comanda il tracciamento dell'asse delle ascisse (asse s)
                                  ' per coordinate ad 1 quadrante

LINE ( 0 , 0 ) - ( 0 , 320 )  ' comanda il tracciamento dell'asse delle ordinate ( asse |M|)
                              ' per coordinate ad 1 quadrante

LOCATE 1,66 : PRINT "MODULO"  ' indicazione della funzione presentata

FOR s = .1 TO 2.5 STEP .001 ' campo di variabilità ed incremento della s per avere 2500 punti di calcolo

x1 = ( 1 / ( 1 + ( s -1/s ) ^ 2 ) ) + 1  ' calcolo della parte reale

y1 = s - ( ( s - 1/s ) / ( 1 + ( s - 1/s ) ^ 2 ) )  ' calcolo della parte immaginaria

M = SQR( x1 ^ 2 + y1 ^ 2 )  ' calcolo del modulo

PSET( 184 * s , 320 - 128 * M ),14  ' presentazione del modulo come una curva continua

NEXT s ' comanda il programma al ritorno automatico all'istruzione FOR s = ecc.

F5
```

si ha la presentazione del grafico dell'andamento del modulo in funzione di s così come mostrato in figura 70. Il modulo ha un massimo per s = .94, un minimo per s = 1.6.

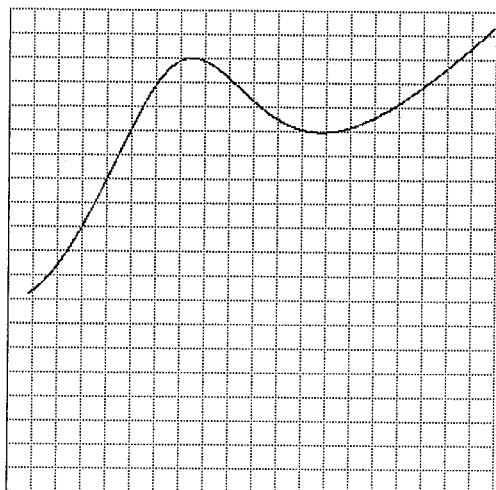



Figura 70
Rappresentazione grafica del modulo dell'ammettenza del sistema risonante

Vediamo infine la presentazione degli argomenti della serie di numeri complessi che deve iniziare dall'esame del grafico globale di figura 67 in cui risulta che tutti i punti sono contenuti nel primo quadrante, ciò dà modo di affermare che gli argomenti stessi sono tutti contenuti entro 90°, si ha perciò:
Per la scala delle ascisse il campo di s dovrà avere gli stessi valori imposti in precedenza:
da s =.1 a s = 2.5.
Dato che l'argomento è al massimo 90°, si adotta anche in questo caso una presentazione ad un solo quadrante, per la quale l'istruzione PSET diventa:

PSET(k1 * s , 320 - k3 * x1) dove k1 = 460 / s = 460 /2.5 = 184
k3 = 320 / Arg° = 320 / 90 = 3.55

Il programma per la presentazione dell'andamento dell'argomento è:

```
LINE ( 0 , 320 ) - ( 460 , 320 )  ' comanda il tracciamento dell'asse delle ascisse (asse s)
                                  ' per coordinate ad 1 quadrante

LINE ( 0 , 0 ) - ( 0 , 320 )      ' comanda il tracciamento dell'asse delle ordinate ( asse Arg°)
                                  ' per coordinate ad 1 quadrante

LOCATE 1,66 : PRINT "ARGOMENTO"   ' indicazione della funzione presentata

FOR s = .1 TO 2.5 STEP .001   ' campo di variabilità ed incremento della s per avere 2500 punti di calcolo

x1 = ( 1 / ( 1 + ( s -1/s ) ^ 2 ) ) + 1   ' calcolo della parte reale

y1 = s - ( ( s - 1/s ) / ( 1 + ( s - 1/s ) ^ 2 ) )   ' calcolo della parte immaginaria            &
```

A = 57.2957 * ATN (y1 / x1) ' calcolo dell'argomento in gradi sessagesimali

PSET(184 * s , 320 - 3.55 * A),14 ' presentazione dell'argomento come una curva continua

NEXT s ' comanda il programma al ritorno automatico all'istruzione FOR s = ecc.

F5

si ha la presentazione del grafico dell'andamento dell'argomento in funzione di s così come mostrato in figura 71. L'argomento ha un massimo per s = .56 ed un minimo per s = 1.2.

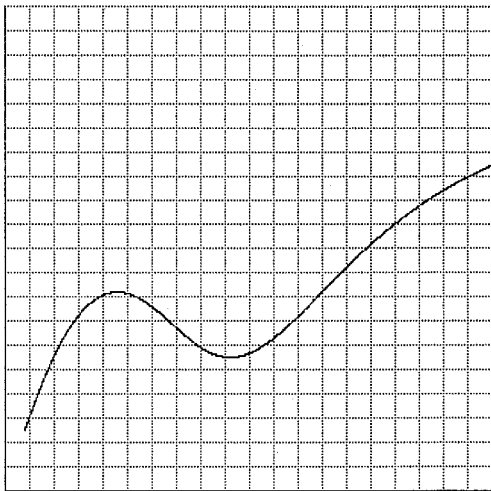



Figura 71
Rappresentazione grafica argomento di numeri complessi funzione di variabile reale

APPENDICE 1
IL QBASIC PER L'ANALISI DEI QUADRIPOLI

Questa appendice è indirizzata agli studenti di elettronica, ai tecnici del ramo e a quanti altri sono interessati alla progettazione dei quadripoli elettrici. Gli esempi di calcolo riportati sono la base per lo sviluppo di più elaborate configurazioni che possono presentarsi nell'ambito delle attività di studio o di lavoro.

A1.1 Applicazione del Qbasic al calcolo dei componenti e della risposta in ampiezza di un filtro Passa Basso.

Il dimensionamento di un filtro passa basso, la cui struttura è riportata in figura 72, è cosa semplice; stabilito infatti il limite della frequenza della banda passante fo ed il valore voluto delle resistenze di terminazione R si calcolano i componenti come segue:

$$L = R / (\pi \, fo)$$

$$C = 1 / (2 \pi \, fo \, R)$$

in cui, espresso fo in Hz ed R in ohm, L è in Henry e C in Farad

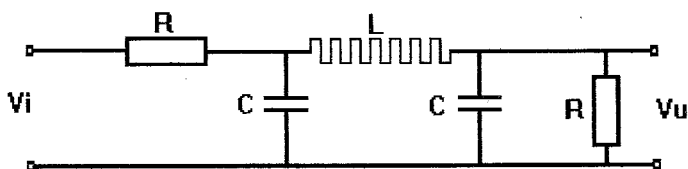



Figura 72
Struttura di un filtro Passa Basso

La determinazione della curva di risposta del filtro Passa Basso, indispensabile per visualizzare l'andamento dell'attenuazione del quadripolo in funzione della frequenza rappresenta, all'opposto del calcolo dei suoi componenti, un problema di notevole difficoltà. Difficoltà superabile con l'impiego della tecnica di calcolo sui numeri complessi sviluppata nel capitolo 11 e con la grafica illustrata nel capitolo 3.

Nel calcolo della risposta del filtro, che imposteremo nella pagina seguente, non si considereranno volutamente le resistenze di perdita dell'induttanza e delle capacità.
Questi elementi, a volte di notevole peso sulla risposta del quadripolo, potranno essere inseriti, come esercizio, nel programma di computazione dal lettore che avrà acquisito l'esperienza necessaria a tale implementazione.

Per calcolare la risposta del quadripolo dobbiamo anzitutto definire in termini complessi i vari componenti che lo costituiscono:
assunto

$w = \omega = 2 \pi f$ la pulsazione angolare

A1 la reattanza dei componenti capacitivi C

C1 la reattanza del componente induttivo L

R il valore delle resistenze di terminazione

si ha

La reattanza (A1) $A1 = 0 - j / w C$

La reattanza (C1) $C1 = 0 + j w L$

La resistenza (R) $R = R + j 0$

Ora, in via del tutto convenzionale, se si assume che il simbolo // indichi il parallelo tra due o più componenti del quadripolo; sulla base della figura 72 (filtro pilotato di tensione) possiamo scrivere:

K per il parallelo tra A1 ed R $K = A1 // R = kx+jky$

B1 per la serie tra C1 e K $B1 = C1 + K = bx+jby$

H per il parallelo tra A1 e B1 $H = A1 // B1 = hx+jhy$

U per la funzione di risposta del filtro $U = ux+juy$

a questo punto mediante passaggi di elettrotecnica classica si arriva alla determinazione della funzione di risposta in frequenza del quadripolo in forma complessa:

$$Vu / Vi = U = (K / B1) / [(R / H) + 1]$$

Per il computo della funzione di risposta si impiegheranno le operazioni tra numeri complessi già trattate nel capitolo 11 con l'aggiunta di una forma di calcolo che ci consentirà l'implementazione in Qbasic del valore complesso del parallelo tra i vari componenti del quadripolo.
Se Q e Z sono ad esempio i valori complessi di due componenti

$Q = qx + j qy$
$Z = zx + j zy$

il loro parallelo sarà:

$$P = Q // Z = (Q \cdot Z) / (Q + Z) = (qx + j qy) \cdot (zx + j zy) / [(qx + j qy) + (zx + j zy)]$$

il valore di P si ottiene quindi applicando in successione le operazioni di "prodotto"; "somma"; "quoziente" tra numeri complessi.
Per lo sviluppo del programma ci serviremo pertanto di 4 Subroutine denominate:

somma: esegue il calcolo (x1+jy1) + (x2+jy2) e fornisce il risultato nella forma (x1 + jy1)

prod: esegue il calcolo (x1+jy1) · (x2+jy2) e fornisce il risultato nella forma (xm + jym)

div: esegue il calcolo (x1+jy1) : (x2+jy2) e fornisce il risultato nella forma (xq + jyq)

parall: esegue il calcolo (x1+jy1) // (x2+jy2) e fornisce il risultato nella forma (xp + jyp)

Per semplificare le procedure di calcolo "parzializzeremo" la funzione U in funzioni più semplici, utilizzando tre variabili complesse di servizio F = fx+jfy; G = gx+jgy; L = lx+jly.
Il programma che ci accingiamo a commentare è diviso in 10 sezioni di lavoro quali:

SEZIONE 1 - impostazione modalità di schermo e richiesta dati (valore della frequenza fo limite della banda passante, valore delle resistenze di terminazione R)

SEZIONE 2 - calcolo dei componenti del filtro (valore dell'induttanza L, valore delle capacità C)

SEZIONE 3 - richiesta dati per il tracciamento della curva di risposta (la frequenza massima Fmax da assegnare alle ascisse del tracciato, il passo -step- per l'incremento di frequenza di calcolo del tracciato)

SEZIONE 4 - formazione del sistema di assi cartesiani ad 1 quadrante (scale delle ascisse in Hz, scale delle ordinate in dB -pari a 2dB/divisione-)

SEZIONE 5 - inizio calcolo automatico in funzione della frequenza

SEZIONE 6 - definizione dei componenti del filtro come numeri complessi

SEZIONE 7 - calcolo di K e di B1

SEZIONE 8 - calcolo di H e della risposta U del filtro in termini complessi

SEZIONE 9 - insieme delle subroutine di calcolo tra numeri complessi che vengono richiamate nel programma

SEZIONE 10 - calcolo del modulo di U e impostazione della funzione grafica PSET

La stesura del programma è la seguente:

```
' SEZIONE 1 - impostazione modalità di schermo e richiesta dati

SCREEN 9

LOCATE 4, 60: PRINT "PASSA BASSO-att."

LOCATE 5, 66: INPUT "fo ="; fo                              &
```

```
LOCATE 6, 66: INPUT "R ="; R

' SEZIONE 2 - calcolo dei componenti del filtro

L = R / (3.14 * fo)

C = 1 / (6.28 * fo * R)

LOCATE 7, 60: PRINT "L="; L

LOCATE 8, 60: PRINT "C="; C

' SEZIONE 3 - richiesta dati per il tracciamento della curva di risposta

LOCATE 9, 66: INPUT "Fmax ="; Fm

LOCATE 10, 66: INPUT "step ="; s

' SEZIONE 4 - formazione del sistema di assi cartesiani ad 1 quadrante

LOCATE 20 , 66: PRINT "y: 2dB/div"

LOCATE 2 , 59 : PRINT "0dB"

LOCATE 24 , 59 : PRINT "- 40dB"

FOR x = 0 TO 460 STEP 23

FOR y = 0 TO 320 STEP 2

PSET (x, y), 7

NEXT y

NEXT x

FOR y = 0 TO 320 STEP 16

FOR x = 0 TO 460 STEP 3

PSET (x, y), 7

NEXT x

NEXT y

LINE (0, 320)-(460, 320)

LINE (0, 0)-(0, 320)

' SEZIONE 5 - inizio calcolo automatico in funzione della frequenza

FOR f = 1 TO Fm STEP s

' SEZIONE 6 - definizione dei componenti del filtro come numeri complessi

' espressione di R = rx+jry                                    &
```

```
rx = R

ry = 0

' espressione di A1 = ax+jay

ax = 0

ay = -1 / (6.28 * f * C)

' espressione di C1 = cx+jcy

cx = 0

cy = (6.28 * f * L)


' SEZIONE 7 - calcolo  di  K e di B1

' espressione di B1 da calcolare B1 = bx+jby

' calcolo di B1= bx+jby = C1+A1//R = C1+ K

' per il computo di K= A1//R = kx+jky si fissa:

x1 = ax

y1 = ay

x2 = rx

y2 = ry

GOSUB parall   ' invio alla  subroutine parall che
               ' esegue il parallelo A1//R ottenendo kx e ky
kx = x1

ky = y1

' per C1 si fissa:

x2 = cx

y2 = cy

GOSUB somma      ' invio alla subroutine somma  che
                 ' esegue la somma  B1 =  C1 + K
bx = x1

by = y1

' si ottiene cosi il valore di B1 = bx+jby


' SEZIONE  8 - calcolo di H e della risposta del filtro in termini complessi

' CALCOLO DI   U= ( K / B1 ) /  [  ( R /  H) + 1  ]                        &
```

```
' 1° -si calcola  H = A1// B1 = hx+jhy

x1 = ax

y1 = ay

x2 = bx

y2 = by

GOSUB parall  '  invio alla subroutine parall che
                 ' esegue il parallelo tra A1 e B1
' si ottiene :

hx = x1

hy = y1

' 2° - si esegue il calcolo F = (R/(A1//B1))+1 =(R/H)+1= fx+jfy
' ( F è la prima variabile di servizio )

x1 = rx

y1 = ry

x2 = hx

y2 = hy

GOSUB div   ' invio alla subroutine div  che esegue il rapporto R/H
               ' (si  esegue direttamente la somma 1+ R/H per ottenere F)
fx = xq + 1

fy = yq

' 3° - si esegue il calcolo 1/F = G = gx+jgy
' ( G è la seconda variabile di servizio )

x1 = 1

y1 = 0

x2 = fx

y2 = fy

GOSUB div  ' invio alla subroutine div  che calcola il reciproco di F

gx = xq

gy = yq

' 4° - si esegue il calcolo L= lx+jly = K/B1
' ( L è la terza variabile di servizio)

x1 = kx

y1 = ky                                                        &
```

```
x2 = bx

y2 = by

GOSUB div   ' invio alla subroutine div che esegue il rapporto K/B l

lx = xq

ly = yq

' 5° -si esegue il prodotto finale  per il calcolo di U  ;  U  = G · L = ux+juy

x1 = gx

y1 = gy

x2 = lx

y2 = ly

GOSUB prod   ' invio alla subroutine prod che esegue il prodotto tra le due
             ' variabili di servizio G ed L

ux = xm

uy = ym

GOTO calcmod  '  ultimato il calcolo di U si passa alla routine
              ' di calcolo per il tracciamento della risposta del filtro


'  SEZIONE 9 - subroutine di calcolo tra numeri complessi che
             ' vengono richiamate dai passi di programma precedenti


' -------SUBROUTINE DI CALCOLO--------
somma:

x1 = x1 + x2

y1 = y1 + y2

RETURN

prod:

xm = (x1 * x2 - y1 * y2)

ym = (x1 * y2 + y1 * x2)

RETURN

div:

xq = (x1 * x2 + y1 * y2) / ((x2) ^ 2 + (y2) ^ 2)

yq = (x2 * y1 - x1 * y2) / ((x2) ^ 2 + (y2) ^ 2)

RETURN                                          &
```

```
parall:

xp = (x1 * x2 - y1 * y2)

yp = (x1 * y2 + y1 * x2)

xs = x1 + x2

ys = y1 + y2

x1 = (xp * xs + yp * ys) / ((xs) ^ 2 + (ys) ^ 2)

y1 = (xs * yp - xp * ys) / ((xs) ^ 2 + (ys) ^ 2)

RETURN


' SEZIONE 10  - calcolo del modulo di U e impostazione
              'della funzione grafica PSET

calcmod :

M = SQR(ux ^ 2 + uy ^ 2)  ' modulo

D = 20 * (LOG(M) / LOG(10))  ' espressione del modulo in dB

PSET ((460 / Fm) * f , -320 / 40 * D) , 14

NEXT f       ' rimanda all'istruzione  For f=1 to Fm
             ' per il calcolo del successivo valore di M
```

A1.2 Esercitazione numerica e grafica per il dimensionamento di un filtro Passa Basso

Sia da calcolare un filtro passa basso con banda limitata alla frequenza fo = 1200 Hz, siano fissati in 1300 ohm i valori delle resistenze di terminazione R, se ne tracci la curva di risposta in ampiezza dalla frequenza 0 alla frequenza Fmax = 2500 Hz con un passo di incremento in frequenza di 10Hz. Impiegando il programma compilato nel paragrafo precedente abbiamo:

F5

FILTRO PASSA BASSO -att.

1ª fase di introduzione dati

```
fo= ? 1200

R= ? 1300

Fmax =? 2500

Step=? 10
```

risultati del calcolo dei componenti
(L in Henry, C in Farad)

```
L = . 3450106

C = 1.020741E-07
```

2^a fase di introduzione dati   Fmax=? 2500

Step=? 10

Dopo l'introduzione dell'ultimo dato si forma il reticolo ad un quadrante ed inizia il tracciamento della curva di risposta, traccia di colore giallo.
Dato che i calcoli per la presentazione della curva di risposta del filtro richiedono molte operazioni, il P.C impiegherà qualche secondo per la visualizzazione grafica completa.
Il tempo di esecuzione del programma dipende naturalmente dal tipo del P.C. e dal valore dell'incremento in frequenza impostato.

Il risultato grafico è riportato in figura 73, da esso si osserva:

-alle frequenze molto inferiori ad fo l'attenuazione del filtro è costante a livello di -6dB, questo valore di attenuazione dipende dal tipo di pilotaggio ipotizzato (pilotaggio di tensione)

-alla frequenza fo il filtro presenta un valore di attenuazione pari a -9dB

-alla frequenza Fmax il filtro presenta un'attenuazione di -25 dB .

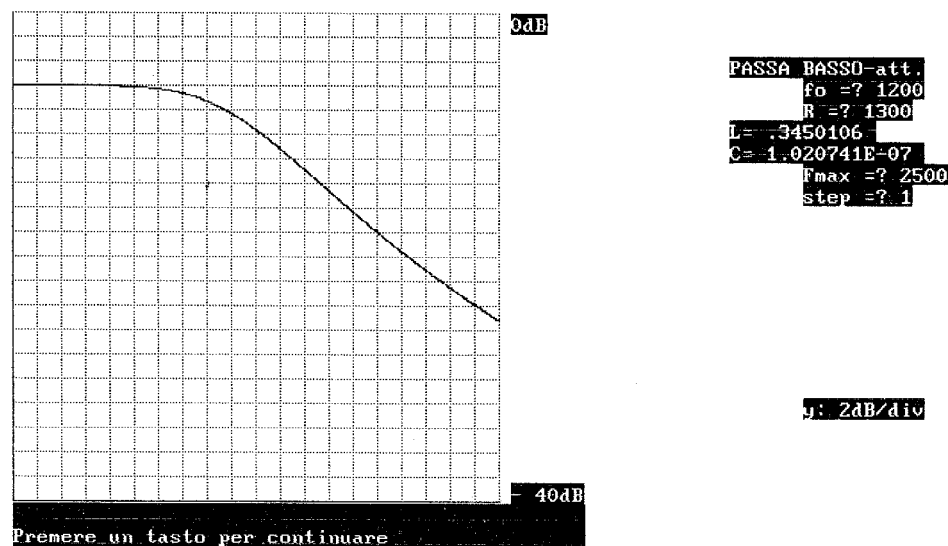



Figura 73
Curva di risposta in ampiezza
del filtro Passa Basso

A1.3 Applicazione del Qbasic al calcolo dei componenti e della risposta in fase di un filtro Passa Basso

Con un programma di elaborazione molto simile a quello mostrato nel paragrafo A1.1 è possibile tracciare l'andamento della risposta in fase del filtro passa basso.
Ferma restando tutta l'impostazione esposta all'inizio del citato paragrafo il nuovo programma si diversifica per il contenuto di alcune delle sezioni di lavoro come segue:

SEZIONE 1 - impostazione modalità di schermo e richiesta dati (valore della frequenza fo limite della banda passante, valore delle resistenze di terminazione R)

SEZIONE 2 - calcolo dei componenti del filtro (valore induttanza L, valore capacità C)

SEZIONE 3 - richiesta dati per il tracciamento della curva di risposta (la frequenza massima Fmax da assegnare alle ascisse del tracciato, il passo -step- per l'incremento di frequenza di calcolo del tracciato)

SEZIONE 4 - formazione del sistema di assi cartesiani ad 1 quadrante (scale delle ascisse in Hz, scale delle ordinate in gradi sessagesimali -pari a 20° /divisione-)

SEZIONE 5 - inizio calcolo automatico in funzione della frequenza

SEZIONE 6 - definizione dei componenti del filtro come numeri complessi

SEZIONE 7 - calcolo di K e di B1

SEZIONE 8 - calcolo di H e della risposta U del filtro in termini complessi

SEZIONE 9 - insieme delle subroutine di calcolo tra numeri complessi che vengono richiamate nel programma

SEZIONE 10 - calcolo dell'argomento Arg e impostazione della funzione grafica PSET

La stesura del programma è la seguente:

```
' SEZIONE 1 - impostazione modalità di schermo e richiesta dati

SCREEN 9

LOCATE 4, 60: PRINT "PASSA BASSO -fase"

LOCATE 5, 66: INPUT "f o ="; fo

LOCATE 6, 66: INPUT "R ="; R

' SEZIONE 2 - calcolo dei componenti del filtro

L = R / (3.14 * fo)

C = 1 / (6.28 * fo * R)

LOCATE 7, 60: PRINT "L="; L

LOCATE 8, 60: PRINT "C="; C

' SEZIONE 3 - richiesta dati per il tracciamento della curva di fase

LOCATE 9, 66: INPUT "Fmax ="; Fm

LOCATE 10, 66: INPUT "step ="; s

' SEZIONE 4 - formazione del sistema di assi cartesiani ad 1 quadrante          &
```

```
LOCATE 20 , 66: PRINT "y: 20°/div"
LOCATE  2 , 59 : PRINT "400°"
LOCATE  24 , 59 : PRINT "0°"
FOR x = 0 TO 460 STEP 23
FOR y = 0 TO 320 STEP 2
PSET (x, y), 7
NEXT y
NEXT x
FOR y = 0 TO 320 STEP 16
FOR x = 0 TO 460 STEP 3
PSET (x, y), 7
NEXT x
NEXT y
LINE (0, 320)-(460, 320)
LINE (0, 0)-(0, 320)
' SEZIONE 5  - inizio calcolo automatico in funzione della frequenza
FOR f = 1 TO  Fm  STEP s

' SEZIONE 6 - definizione dei componenti del filtro come numeri complessi
' espressione di R = rx+jry
rx = R
ry = 0
' espressione di A1 = ax+jay
ax = 0
ay = -1 / (6.28 * f * C)
' espressione di C1 = cx+jcy
cx = 0
cy = (6.28 * f * L)

' SEZIONE 7 - calcolo  di  K e di B1
' espressione di B1 da calcolare B1 = bx+jby                              &
```

```
' calcolo di B1= bx+jby = C1+A1//R = C1+ K

' per il computo di K= A1//R = kx+jky si fissa:

x1 = ax

y1 = ay

x2 = rx

y2 = ry

GOSUB parall   ' invio alla subroutine parall che
               ' esegue il parallelo A1//R ottenendo kx e ky
kx = x1

ky = y1

' per C1 si fissa:

x2 = cx

y2 = cy

GOSUB somma     ' invio alla subroutine somma che
                ' esegue la somma B1 = C1 + K

bx = x1

by = y1

' si ottiene cosi il valore di B1 = bx+jby


' SEZIONE 8 - calcolo di H e della risposta del filtro in termini complessi

' CALCOLO DI   U= ( K / B1 ) / [ ( R / H ) + 1 ]

' 1° -si calcola H = A1// B1 = hx+jhy

x1 = ax

y1 = ay

x2 = bx

y2 = by

GOSUB parall  ' invio alla subroutine parall che
              ' esegue il parallelo tra A1 e B1
' si ottiene :

hx = x1

hy = y1


' 2° - si esegue il calcolo F = (R/(A1//B1))+1 =(R/H)+1= fx+jfy
' ( F è la prima variabile di servizio )                                &
```

```
x1 = rx

y1 = ry

x2 = hx

y2 = hy

GOSUB div   ' invio alla subroutine div  che esegue il rapporto R/H
            ' (si  esegue direttamente la somma 1+ R/H per ottenere F)
fx = xq + 1

fy = yq


' 3° - si esegue il calcolo 1/F = G = gx+jgy
' ( G è la seconda variabile di servizio )

x1 = 1

y1 = 0

x2 = fx

y2 = fy

GOSUB div   ' invio alla subroutine div  che calcola il reciproco di F

gx = xq

gy = yq


' 4° - si esegue il calcolo L= lx+jly = K/B1
' ( L è la terza variabile di servizio)

x1 = kx

y1 = ky

x2 = bx

y2 = by

GOSUB div  ' invio alla subroutine div che esegue il rapporto K/B1

lx = xq

ly = yq


' 5° -si esegue il prodotto finale  per il calcolo di U  ;  U  = G · L = ux+juy

x1 = gx

y1 = gy

x2 = lx

y2 = ly
```

&

```
GOSUB prod  ' invio alla subroutine prod che esegue il prodotto tra le due
            ' variabili di servizio G ed L
ux = xm

uy = ym

GOTO calcarg  ' ultimato il calcolo di U si passa alla routine
              ' di calcolo per il tracciamento della risposta in fase del filtro


' SEZIONE 9 - subroutine di calcolo tra numeri complessi che
            ' vengono richiamate dai passi di programma precedenti

'-------SUBROUTINE DI CALCOLO--------
somma:

x1 = x1 + x2

y1 = y1 + y2

RETURN

prod:

xm = (x1 * x2 - y1 * y2)

ym = (x1 * y2 + y1 * x2)

RETURN

div:

xq = (x1 * x2 + y1 * y2) / ((x2) ^ 2 + (y2) ^ 2)

yq = (x2 * y1 - x1 * y2) / ((x2) ^ 2 + (y2) ^ 2)

RETURN

parall:

xp = (x1 * x2 - y1 * y2)

yp = (x1 * y2 + y1 * x2)

xs = x1 + x2

ys = y1 + y2

x1 = (xp * xs + yp * ys) / ((xs) ^ 2 + (ys) ^ 2)

y1 = (xs * yp - xp * ys) / ((xs) ^ 2 + (ys) ^ 2)

RETURN


' SEZIONE 10  - calcolo dell'argomento di U e impostazione
              ' della funzione grafica PSET

calcarg :                                              &
```

```
IF (ux = 0) AND (uy <> 0) THEN R = .00000001#

IF (ux > 0) AND (uy = 0) THEN T = 0

IF (ux < 0) AND (uy = 0) THEN T = 180

IF (uy > 0) AND (ux > 0) THEN T = 0

IF (uy > 0) AND (ux < 0) THEN T = 180

IF (uy < 0) AND (ux < 0) THEN T = 180

IF (uy < 0) AND (ux > 0) THEN T = 360

arg = T + 57.2957 * ATN(uy / ux)

PSET ((460 / Fm) * f, 320 - .8 * (360 - arg)), 14

NEXT f      ' rimanda all'istruzione For f=1 to Fm
            ' per il calcolo del successivo valore di Arg
```

A1.4 Esercitazione numerica e grafica per il tracciamento della risposta in fase di un filtro Passa Basso

Sia da tracciare la risposta in fase del filtro passa basso con banda limitata alla frequenza fo = 1200 Hz, siano fissati in 1300 ohm i valori delle resistenze di terminazione R, se ne tracci la curva di risposta dalla frequenza 0 alla frequenza Fmax = 2500 Hz con un passo di incremento in frequenza di 10Hz.

Impiegando il programma compilato nel paragrafo precedente abbiamo:

F5

FILTRO PASSA BASSO-fase

1ª fase di introduzione dati

fo= ? 1200

R= ? 1300

Fmax =? 2500

Step=? 10

risultati del calcolo dei componenti
(L in Henry, C in Farad)

L = . 3450106

C = 1.020741E-07

2ª fase di introduzione dati

Fmax=? 2500

Step=? 10

Dopo l'introduzione dell'ultimo dato si forma il reticolo ad un quadrante ed inizia il tracciamento della curva di risposta in fase, traccia di colore giallo.
Il risultato grafico è riportato in figura 74, da esso si osserva che la curva di fase del filtro passa basso cresce con la frequenza con legge non lineare, la variazione di fase è contenuta tra 0° e 210° circa.

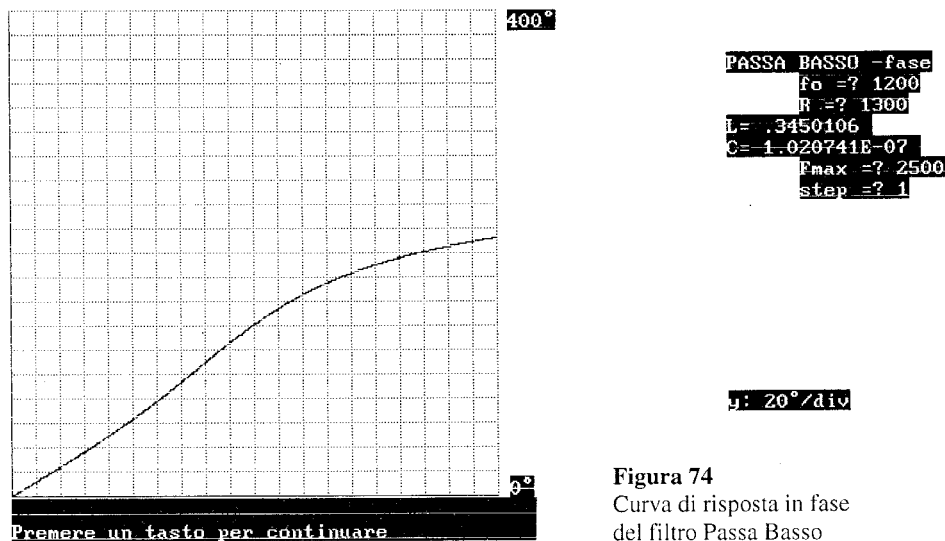



Figura 74
Curva di risposta in fase del filtro Passa Basso

A1.5 Applicazione del Qbasic al calcolo dei componenti e della risposta in ampiezza di un filtro Passa Banda

Per il dimensionamento di un filtro passa banda, la cui struttura è riportata in figura 75, si fissano i limiti di banda passante F1 ed F2 ed il valore voluto delle resistenze di terminazione R, si calcolano i componenti come segue:

$$L1 = R / [\pi (F2 - F1)]$$

$$L2 = R (F2 - F1) / (2 \pi F1 F2)$$

$$C1 = (F2 - F1) / (4 \pi F1 F2 R)$$

$$C2 = 1 / [2 \pi (F2 - F1) R]$$

in cui, espresse F1 ed F2 in Hz ed R in ohm, L1 ed L2 sono in Henry e C1 e C2 in Farad

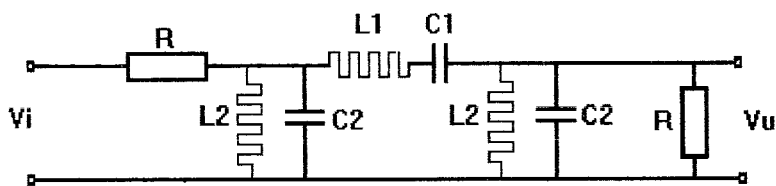



Figura 75
Struttura di un filtro Passa Banda

Per calcolare la risposta del quadripolo dobbiamo anzitutto definire, in termini complessi, i vari componenti che lo costituiscono:
assunto

$w = \omega = 2 \pi f$ la pulsazione angolare

si ha

La reattanza di L1 = l1x+j l1y = $0 + j\, 2 \pi f$ L1

La reattanza di L2 = l2x+j l2y = $0 + j\, 2 \pi f$ L2

La reattanza di C1 = c1x+j c1y = $0 - j / 2 \pi f$ C1

La reattanza di C2 = c2x+j c2y = $0 - j / 2 \pi f$ C2

La resistenza R = R + j 0

sulla base della figura 75 (filtro pilotato di tensione) possiamo scrivere:

per il parallelo tra L1 e C2 A = L1 // C2 = ax + j ay

per la serie tra L1 e C1 B = L1 + C1 = bx + j by

per il parallelo tra A ed R C = A // R = cx + j cy

per la serie tra B e C D = B + C = dx + j dy

per il parallelo tra A e D E = A // D = ex + j ey

per la serie tra R ed E F = R + E = fx + j fy

per il reciproco di F G = 1 / F = gx + j gy

per il prodotto tra G ed E $H = G \cdot E = hx + j\, hy$

per il prodotto tra H e C $I = H \cdot C = ix + j\, iy$

da cui la funzione di risposta:

$$Vu / Vi = U = I / D = ux + j\, uy$$

Il programma per il calcolo ed il tracciamento della curva di risposta in ampiezza del filtro passa banda è diviso in 10 sezioni:

SEZIONE 1 - impostazione modalità di schermo e richiesta dati (valori delle frequenze F1 ed F2 limiti della banda passante, valore delle resistenze di terminazione R)

SEZIONE 2 - calcolo dei componenti del filtro (valori delle induttanze L1, L2; valori delle capacità C1, C2)

SEZIONE 3 - richiesta dati per il tracciamento della curva di risposta (la frequenza minima Fmin e la frequenza massima Fmax da assegnare alle ascisse del tracciato, il passo -step- per l'incremento di frequenza di calcolo del tracciato)

SEZIONE 4 - formazione del sistema di assi cartesiani ad 1 quadrante (scale delle ascisse in Hz, scale delle ordinate in dB -pari a 2dB/divisione-)

SEZIONE 5 - inizio calcolo automatico in funzione della frequenza

SEZIONE 6 - definizione dei componenti del filtro come numeri complessi

SEZIONE 7 - calcolo delle variabili complesse A; B; C; D; E; F; G; H; I

SEZIONE 8 - calcolo della risposta U del filtro in termini complessi

SEZIONE 9 - insieme delle subroutine di calcolo tra numeri complessi che vengono richiamate nel programma

SEZIONE 10 - calcolo del modulo di U e impostazione della funzione grafica PSET

Viene di seguito compilato e commentato il programma per il calcolo dei componenti ed il tracciamento della curva di risposta in ampiezza del filtro passa banda:

```
' SEZIONE 1 - impostazione modalità di schermo e richiesta dati

SCREEN 9

LOCATE 4, 60: PRINT "PASSA BANDA-att."

LOCATE 5, 66: INPUT "F1="; F1

LOCATE 6, 66: INPUT "F2="; F2

LOCATE 7, 66: INPUT "R="; R                    &
```

```
' SEZIONE 2 - calcolo dei componenti del filtro

L1 = R / (3.14 * (F2 - F1))

L2 = R * (F2 - F1) / (6.28 * F1 * F2)

C1 = (F2 - F1) / (12.56 * F1 * F2 * R)

C2 = 1 / (6.28 * (F2 - F1) * R)

LOCATE 8, 60: PRINT "L1="; L1

LOCATE 9, 60: PRINT "L2="; L2

LOCATE 10, 60: PRINT "C1="; C1

LOCATE 11, 60: PRINT "C2="; C2

' SEZIONE 3 - richiesta dati per il tracciamento della curva di risposta

LOCATE 12, 66: INPUT "Fmin="; Fi

LOCATE 13, 66: INPUT "Fmax="; Fm

LOCATE 14, 66: INPUT "step="; s

' SEZIONE 4 - formazione del sistema di assi cartesiani ad 1 quadrante

LOCATE 20, 66: PRINT "y: 2dB/div"

LOCATE 2 , 59 : PRINT " 0 dB"

LOCATE 24 , 59 : PRINT"-40dB"

FOR x = 0 TO 460 STEP 23

FOR y = 0 TO 320 STEP 2

PSET (x, y), 7

NEXT y

NEXT x

FOR y = 0 TO 320 STEP 16

FOR x = 0 TO 460 STEP 3

PSET (x, y), 7

NEXT x

NEXT y

LINE (0, 320)-(460, 320)

LINE (0, 0)-(0, 320)                                    &
```

' SEZIONE 5 - inizio calcolo automatico in funzione della frequenza

FOR F = (Fi + 1) TO Fm STEP s

' SEZIONE 6 - definizione dei componenti del filtro come numeri complessi
' espressione di R = rx+jry

rx = R

ry = 0

' espressione di L1 =l1x+jl1y

l1x = 0

l1y = 6.28 * F * L1

' espressione di L2 = l2x,+jly

l2x = 0

l2y = 6.28 * F * L2

' espressione di C1 = c1x+jc1y

c1x = 0

c1y = -1 / (6.28 * F * C1)

' espressione di C2 = c2x+jc2y

c2x = 0

c2y = -1 / (6.28 * F * C2)

' SEZIONE 7 - calcolo di A,B,C,D,E,F,G,H,I in termini complessi

' computo di A = L2//C2 = ax+jay

x1 = l2x

y1 = l2y

x2 = c2x

y2 = c2y

GOSUB parall ' invia alla subroutine parall per il calcolo di A

ax = x1

ay = y1

' computo di B = L1+C1 =bx+jby

x1 = l1x &

```
y1 = l1y
x2 = c1x
y2 = c1y
GOSUB somma  ' invia alla subroutine somma per il calcolo di B
bx = x1
by = y1
' computo di C = A//R = cx+jcy
x1 = ax
y1 = ay
x2 = rx
y2 = ry
GOSUB parall  ' invia alla subroutine parall  per il calcolo di C
cx = x1
cy = y1
' computo di D = B+C =dx+jdy
x1 = bx
y1 = by
x2 = cx
y2 = cy
GOSUB somma   '  invia alla subroutine somma per il calcolo di D
dx = x1
dy = y1
' computo di E = A//D = ex+jey
x1 = ax
y1 = ay
x2 = dx
y2 = dy
GOSUB parall  ' invia alla subroutine parall per il calcolo di E
ex = x1                                                      &
```

ey = y1

' computo di F = R+E =fx+jfy

x1 = rx

y1 = ry

x2 = ex

y2 = ey

GOSUB somma ' invia alla subroutine somma per il calcolo di F

fx = x1

fy = y1

' computo di G = 1/F =gx+jgy

x1 = 1

y1 = 0

x2 = fx

y2 = fy

GOSUB div ' invia alla subroutine div per il calcolo di G

gx = xq

gy = yq

' computo di H = G·E =hx+jhy

x1 = gx

y1 = gy

x2 = ex

y2 = ey

GOSUB prod ' invia alla subroutine prod per il calcolo di H

hx = xm

hy = ym

' computo di I = H·C =ix+jiy

x1 = hx

y1 = hy

x2 = cx &

```
y2 = cy

GOSUB prod   ' invia alla subroutine prod per il calcolo di I

ix = xm

iy = ym

' SEZIONE 8 calcolo della risposta del filtro

' computo di U = I/D = ux+juy

x1 = ix

y1 = iy

x2 = dx

y2 = dy

GOSUB div   ' invia alla subroutine div  per il calcolo di U

ux = xq

uy = yq

GOTO calcom   ' invia alla routine  calcom per il calcolo del modulo di U

' SEZIONE  9

'-------SUBROUTINE DI CALCOLO---

somma:

x1 = x1 + x2

y1 = y1 + y2

RETURN

prod:

xm = (x1 * x2 - y1 * y2)

ym = (x1 * y2 + y1 * x2)

RETURN

div:

xq = (x1 * x2 + y1 * y2) / ((x2) ^ 2 + (y2) ^ 2)

yq = (x2 * y1 - x1 * y2) / ((x2) ^ 2 + (y2) ^ 2)

RETURN

parall:                                                    &
```

```
xp = (x1 * x2 - y1 * y2)

yp = (x1 * y2 + y1 * x2)

xs = x1 + x2

ys = y1 + y2

x1 = (xp * xs + yp * ys) / ((xs) ^ 2 + (ys) ^ 2)

y1 = (xs * yp - xp * ys) / ((xs) ^ 2 + (ys) ^ 2)

RETURN

' SEZIONE 10 calcolo del modulo e impostazione della funzione PSET

'calcolo del modulo

calcom:

M = SQR(ux ^ 2 + uy ^ 2)    ' calcolo del  modulo

D = 20 * (LOG(M) / LOG(10))  ' espressione del modulo in dB

PSET ((460 / Fm) * F, -320 / 40 * D), 14

NEXT F       ' rimanda all'istruzione  For F= .....
             ' per il calcolo del successivo valore di M
```

A1.6 Esercitazione numerica e grafica per il tracciamento della risposta in ampiezza di un filtro Passa Banda

Sia da tracciare la risposta in ampiezza di un filtro passa banda con banda limitata tra F1 = 5000 Hz e F2 = 10000 Hz, siano fissati in 2000 ohm i valori delle resistenze di terminazione R, se ne tracci la curva di risposta dalla frequenza Fmin =1000 Hz alla frequenza Fmax = 20000 Hz con un passo di incremento in frequenza di 50Hz.

Impiegando il programma compilato nel paragrafo precedente abbiamo:

F5

FILTRO PASSA BANDA-att.

1ª fase di introduzione dati

F1= ? 5000

F2=? 10000

R= ? 2000

risultati del calcolo dei componenti
(L in Henry, C in Farad)

L1 = .1273885

L2 = 3.184713E-02

C1 = 3.980892E-09

C2 = 1.592357E-08

2ª fase di introduzione dati Fmin=? 1000

Fmax=? 20000

Step=? 50

Dopo l'introduzione dell'ultimo dato si forma il reticolo ad un quadrante ed inizia il tracciamento della curva di risposta in ampiezza, traccia di colore giallo.

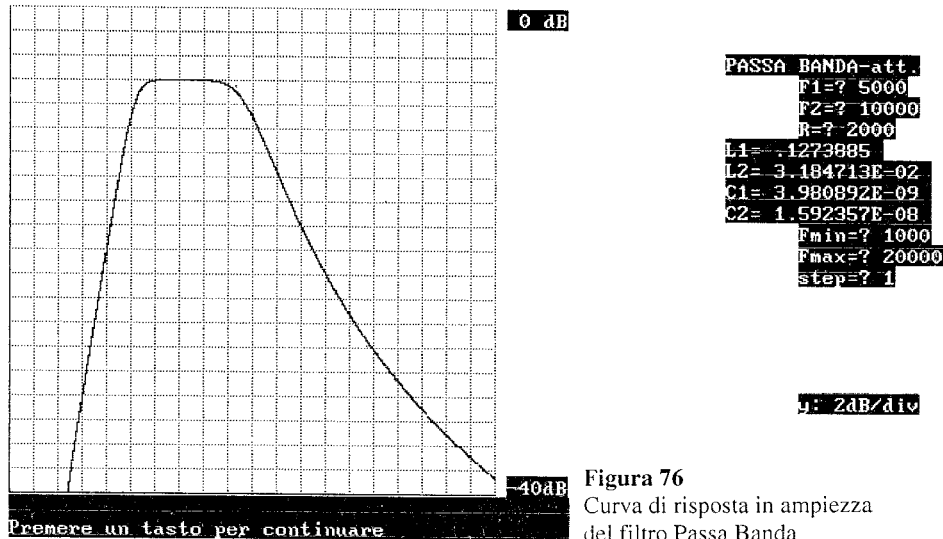



Figura 76
Curva di risposta in ampiezza del filtro Passa Banda

Il risultato grafico è riportato in figura 76, da esso si osserva:

-alle frequenze F1=5000 Hz e F2=10000 Hz l'attenuazione del filtro è di -9dB

-all'interno di questo intervallo di frequenza si ha una attenuazione costante a livello di -6dB

-alla frequenza F= 3300 Hz il filtro presenta un valore di attenuazione pari a -40dB

-alla frequenza F= 20000 Hz il filtro presenta un'attenuazione di - 39dB .

A1.7 Applicazione del Qbasic al calcolo dei componenti e della risposta in fase di un filtro Passa Banda

Con un programma di elaborazione simile a quello mostrato nel paragrafo A1.5 è possibile tracciare l'andamento della risposta in fase del filtro passa banda.
Il contenuto delle sezioni di lavoro varia come segue:

SEZIONE 1 - impostazione modalità di schermo e richiesta dati (valori delle frequenze F1 ed F2 limiti della banda passante, valore delle resistenze di terminazione R)

SEZIONE 2 - calcolo dei componenti del filtro (valori delle induttanze L1, L2; valori delle capacità C1, C2)

SEZIONE 3 - richiesta dati per il tracciamento della curva di fase (la frequenza minima Fmin e la frequenza massima Fmax da assegnare alle ascisse del tracciato, il passo -step- per l'incremento di frequenza di calcolo del tracciato)

SEZIONE 4 - formazione del sistema di assi cartesiani ad 1 quadrante (scale delle ascisse in Hz, scale delle ordinate in gradi sessagesimale -pari a 20° /divisione-)

SEZIONE 5 - inizio calcolo automatico in funzione della frequenza

SEZIONE 6 - definizione dei componenti del filtro come numeri complessi

SEZIONE 7 - calcolo delle variabili complesse A; B; C; D; E; F; G; H; I

SEZIONE 8 - calcolo della risposta U del filtro in termini complessi

SEZIONE 9 - insieme delle subroutine di calcolo tra numeri complessi che vengono richiamate nel programma

SEZIONE 10 - calcolo dell'argomento Arg di U e impostazione della funzione grafica PSET

La stesura del programma è la seguente:

```
' SEZIONE 1 - impostazione modalità di schermo e richiesta dati

SCREEN 9

LOCATE 4, 60: PRINT "PASSA BANDA-fase"

LOCATE 5, 66: INPUT "F1="; F1

LOCATE 6, 66: INPUT "F2="; F2

LOCATE 7, 66: INPUT "R="; R

' SEZIONE 2 - calcolo dei componenti del filtro

L1 = R / (3.14 * (F2 - F1))

L2 = R * (F2 - F1) / (6.28 * F1 * F2)

C1 = (F2 - F1) / (12.56 * F1 * F2 * R)

C2 = 1 / (6.28 * (F2 - F1) * R)

LOCATE 8, 60: PRINT "L1="; L1

LOCATE 9, 60: PRINT "L2="; L2

LOCATE 10, 60: PRINT "C1="; C1

LOCATE 11, 60: PRINT "C2="; C2

' SEZIONE 3 - richiesta dati per il tracciamento della curva di fase

LOCATE 12, 66: INPUT "Fmin="; Fi                              &
```

```
LOCATE 13, 66: INPUT "Fmax="; Fm

LOCATE 14, 66: INPUT "step="; s

' SEZIONE 4 - formazione del sistema di assi cartesiani ad 1 quadrante

LOCATE 20, 66: PRINT "y: 20°/div"

LOCATE 2 , 59 : PRINT " 400°"

LOCATE 24 , 59 : PRINT"0°"

FOR x = 0 TO 460 STEP 23

FOR y = 0 TO 320 STEP 2

PSET (x, y), 7

NEXT y

NEXT x

FOR y = 0 TO 320 STEP 16

FOR x = 0 TO 460 STEP 3

PSET (x, y), 7

NEXT x

NEXT y

LINE (0, 320)-(460, 320)

LINE (0, 0)-(0, 320)

' SEZIONE 5 - inizio calcolo automatico in funzione della frequenza

FOR F = (Fi + 1) TO Fm STEP s

' SEZIONE 6 - definizione dei componenti del filtro come numeri complessi
' espressione di R = rx+jry

rx = R

ry = 0

' espressione di L1 =l1x+jl1y

l1x = 0

l1y = 6.28 * F * L1

' espressione di L2 = l2x,+jly

l2x = 0
```

&

```
l2y = 6.28 * F * L2
' espressione di C1 = c1x+jc1y
c1x = 0
c1y = -1 / (6.28 * F * C1)
' espressione di C2 = c2x+jc2y
c2x = 0
c2y = -1 / (6.28 * F * C2)

' SEZIONE 7 - calcolo  di A,B,C,D,E,F,G,H,I in termini complessi

' computo di A = L2//C2 = ax+jay
x1 = l2x
y1 = l2y
x2 = c2x
y2 = c2y
GOSUB parall  ' invia alla subroutine parall  per il calcolo di A
ax = x1
ay = y1
' computo di B = L1+C1 =bx+jby
x1 = l1x
y1 = l1y
x2 = c1x
y2 = c1y
GOSUB somma  ' invia alla subroutine somma per il calcolo di B
bx = x1
by = y1
' computo di C = A//R = cx+jcy
x1 = ax
y1 = ay
x2 = rx
y2 = ry                                                  &
```

```
GOSUB parall  ' invia alla subroutine parall  per il calcolo di C
cx = x1
cy = y1

' computo di D = B+C =dx+jdy
x1 = bx
y1 = by
x2 = cx
y2 = cy
GOSUB somma  ' invia alla subroutine somma per il calcolo di D
dx = x1
dy = y1
' computo di E = A//D = ex+jey
x1 = ax
y1 = ay
x2 = dx
y2 = dy
GOSUB parall  ' invia alla subroutine parall per il calcolo di E
ex = x1
ey = y1
' computo di F = R+E =fx+jfy
x1 = rx
y1 = ry
x2 = ex
y2 = ey
GOSUB somma  ' invia alla subroutine somma per il calcolo di  F
fx = x1
fy = y1

' computo di G = 1/F =gx+jgy
x1 = 1                                                  &
```

```
y1 = 0
x2 = fx
y2 = fy
GOSUB div   ' invia alla subroutine div per il calcolo di G
gx = xq
gy = yq

' computo di H = G·E =hx+jhy
x1 = gx
y1 = gy
x2 = ex
y2 = ey
GOSUB prod   ' invia alla subroutine prod per il calcolo di H
hx = xm
hy = ym
' computo di I = H·C =ix+jiy
x1 = hx
y1 = hy
x2 = cx
y2 = cy
GOSUB prod   ' invia alla subroutine prod per il calcolo di I
ix = xm
iy = ym

' SEZIONE 8 calcolo della risposta del filtro
 ' computo di U = I/D = ux+juy
x1 = ix
y1 = iy
x2 = dx
y2 = dy
GOSUB div   ' invia alla subroutine div  per il calcolo di U                    &
```

```
ux = xq
uy = yq
GOTO calcarg  ' invia alla routine  calcarg per il calcolo dell'argomento Arg di U
' SEZIONE 9

'-------SUBROUTINE DI CALCOLO--------
somma:
x1 = x1 + x2
y1 = y1 + y2
RETURN
prod:
xm = (x1 * x2 - y1 * y2)
ym = (x1 * y2 + y1 * x2)
RETURN
div:
xq = (x1 * x2 + y1 * y2) / ((x2) ^ 2 + (y2) ^ 2)
yq = (x2 * y1 - x1 * y2) / ((x2) ^ 2 + (y2) ^ 2)
RETURN
parall:
xp = (x1 * x2 - y1 * y2)
yp = (x1 * y2 + y1 * x2)
xs = x1 + x2
ys = y1 + y2
x1 = (xp * xs + yp * ys) / ((xs) ^ 2 + (ys) ^ 2)
y1 = (xs * yp - xp * ys) / ((xs) ^ 2 + (ys) ^ 2)
RETURN
' SEZIONE 10 calcolo dell'argomento e impostazione della funzione PSET
' calcolo dell'argomento
calcarg:
IF (ux = 0) AND (uy <> 0) THEN R = .00000001#                              &
```

```
IF (ux > 0) AND (uy = 0) THEN T = 0

IF (ux < 0) AND (uy = 0) THEN T = 180

IF (uy > 0) AND (ux > 0) THEN T = 0

IF (uy > 0) AND (ux < 0) THEN T = 180

IF (uy < 0) AND (ux < 0) THEN T = 180

IF (uy < 0) AND (ux > 0) THEN T = 360

arg = T + 57.2957 * ATN(uy / ux)

PSET ((460 / Fm) * f, 320 - .8 * (360 - arg)), 14

NEXT F      ' rimanda all'istruzione For F= .....
            ' per il calcolo del successivo valore di Arg
```

A1.8 Esercitazione numerica e grafica per il tracciamento della risposta in fase di un filtro Passa Banda

Sia da tracciare la risposta in fase di un filtro passa banda con banda limitata tra F1 = 5000 Hz e F2 = 10000 Hz, siano fissati in 2000 ohm i valori delle resistenze di terminazione R, se ne tracci la curva di fase dalla frequenza Fmin = 1000 Hz alla frequenza Fmax = 20000 Hz con un passo di incremento in frequenza di 50 Hz.

Impiegando il programma compilato nel paragrafo precedente abbiamo:

F5

FILTRO PASSA BANDA-fase

1ª fase di introduzione dati

F1=? 5000

F2=? 10000

R= ? 2000

risultati del calcolo dei componenti
(L in Henry, C in Farad)

L1 = .1273885

L2 = 3.184713E-02

C1 = 3.980892E-09

C2 = 1.592357E-08

2ª fase di introduzione dati

Fmin=? 1000

Fmax=? 20000

Step=? 50

Dopo l'introduzione dell'ultimo dato si forma il reticolo ad un quadrante ed inizia il tracciamento della curva di risposta in fase mostrata in figura 77, traccia di colore giallo.
Dalla curva si osserva che il tracciato presenta una apparente discontinuità alla frequenza media geometrica tra F1 ed F2 (7071 Hz); tale fenomeno non rappresenta una reale discontinuità ma è il naturale passaggio tra il valore della fase che ha raggiunto 360° e la ripresa di crescita della fase stessa che inizia da 0°. Infatti il primo tratto di curva si estende da 100° alla frequenza di 1000 Hz fino a 360° raggiunti alla frequenza di 7071 Hz. Il secondo tratto riprende con 0° alla frequenza di 7071 Hz fino a circa 238° alla frequenza di 20000 Hz.

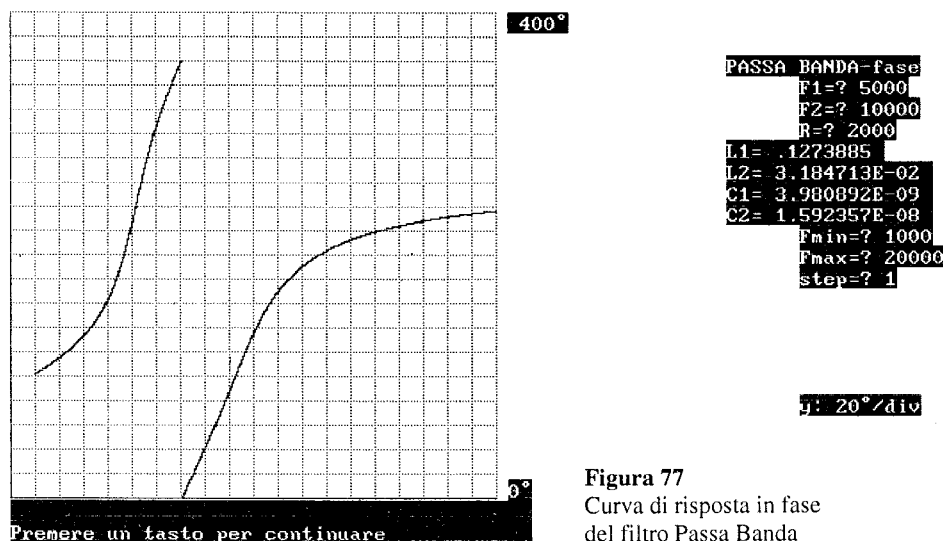



Figura 77
Curva di risposta in fase
del filtro Passa Banda

A1.9 Conclusioni

Gli esercizi svolti in questa appendice sono esempi di ciò che si può fare implementando in Qbasic le metodologie di calcolo dei numeri complessi per l'analisi dei quadripoli.
Impiegando il metodo di "parzializzazione" per isolare le diverse componenti reattive delle reti, similmente a quanto fatto ad esempio per ottenere le A; B; C; D; E; F; G; H; I dalla struttura passa banda di figura 75, è possibile affrontare lo studio di quadripoli di notevole complessità.
Seguendo le modalità di compilazione dei programmi mostrati in precedenza, ampliandoli e modificandoli per adattarli alle nuove problematiche, si potranno ottenere importanti risultati.

APPENDICE 2

LA SOLUZIONE DI UN PROBLEMA DI GEOMETRIA ANALITICA

Nella presente appendice viene illustrato un metodo di implementazione in Qbasic dei processi di calcolo per la soluzione di un problema caratteristico di geometria analitica.

A2.1 Sui problemi di geometria analitica

La tematica relativa ai problemi di geometria analitica è molto vasta. Innumerevoli sono gli esercizi che si possono proporre, ciascuno con un particolare grado di difficoltà. Non è possibile pertanto spaziare su questo argomento in una ridotta appendice del testo.
Ci limiteremo perciò a prendere in esame la soluzione di un esercizio di media complessità nell'intento di mostrare al lettore le procedure generali per arrivare all'implementazione in Qbasic degli algoritmi che sviluppano il calcolo numerico e la presentazione grafica che risolve il problema proposto.

A2.2 Proposizione del problema

Si determinino le equazioni delle rette passanti per un punto Po e tangenti ad una circonferenza C avente il centro nell'origine degli assi.
Si calcolino le coordinate dei punti di tangenza individuando l'equazione della retta passante per detti punti.
Si tracci infine il grafico risolutivo in un sistema di assi cartesiani calibrato.

A2.3 Soluzione analitica del problema

La soluzione analitica del problema prevede inizialmente l'impostazione dell'equazione generale di una retta passante per un punto Po di coordinate (xo ; yo); l'equazione della retta è:

$$(y - yo) / (x - xo) = m$$

da cui

$$y = m x - m xo + yo$$

dove **m** è il coefficiente angolare
$(- m xo + yo) =$ **n** è il parametro

La retta così specificata deve essere messa a sistema con l'equazione della circonferenza in cui si cercano i punti di tangenza:

$$\begin{cases} y = m x - m xo + yo \\ y^2 + x^2 = R^2 \end{cases}$$

Si risolve il sistema in x ponendo il secondo membro della prima equazione al posto della variabile dipendente y dell'equazione della circonferenza; si ha:

$$(m x - m xo + yo)^2 + x^2 = R^2$$

sviluppando si ottiene l'equazione di secondo grado in x:

$$x^2 (m^2 + 1) - 2\, m\, x\, (m\, xo - yo) - R^2 + (m\, xo - yo)^2 = 0$$

Sappiamo che per avere la condizione di tangenza tra le rette e la circonferenza, l'equazione trovata deve avere radici reali e coincidenti; cioè deve essere nullo il discriminante D dell'equazione:

$$D = b^2 - 4\, a\, c = 0$$

dove

$$a = (m^2 + 1)$$
$$b = - 2\, m\, (m\, xo - yo)$$
$$c = - R^2 + (m\, xo - yo)^2$$

perciò si deve scrivere:

$$[- 2\, m\, (m\, xo - yo)]^2 - 4\, (m^2 + 1)\, [-R^2 + (m\, xo - yo)^2] = 0$$

la nuova equazione in m, sviluppata e risolta dà:

$$m = \left\{ - (2\, xo\, yo) \pm [(- 2\, xo\, yo)^2 - 4\, (R^2 - xo^2)\, (R^2 - yo^2)]^{1/2} \right\} / 2\, (R^2 - xo^2)$$

Come era previsto l'equazione fornisce due valori di m: m1; m2, sono infatti due le rette tangenti alla circonferenza data; i valori di m soddisferanno il problema soltanto se il punto Po sarà esterno alla circonferenza. Ciò è ovvio, altrimenti le rette sarebbero secanti e non sussisterebbe la condizione imposta $D = 0$.
Deve pertanto essere sempre verificata una delle seguenti disuguaglianze:

$$|xo| > R$$

$$|yo| \geq R$$

Il problema non ha soluzione numerica se $|xo| = R$ dato che si annulla il denominatore dell'espressione in m creando una condizione di infinito. In questa situazione il calcolo algebrico di m non è possibile; è il caso di una retta tangente alla circonferenza e parallela all'asse y.
L'espressione in m alla quale siamo pervenuti dovrà essere implementata in Qbasic per il calcolo dei valori di m1 ed m2 che interessano il nostro problema.
Il tracciamento delle rette tangenti alla circonferenza presuppone anche il calcolo dei valori di n1 ed n2 secondo l'equazione

$$n = yo - xo\, m$$

anche questa espressione dovrà essere implementata nella routine di programma.
Il problema si completerà con la determinazione delle coordinate (x1 ; y1) e (x2 ; y2) relative ai due punti, P1 e P2, di tangenza sulla circonferenza e con la stesura dell'equazione della retta passante per questi.
Le coordinate dei due punti sono le soluzioni del sistema iniziale dopo che in esso sono stati inseriti i valori calcolati di (m1 ; n1) e (m2 ; n2):

$$\begin{cases} y = m1\, x + n1 \\ y^2 + x^2 = R^2 \end{cases} \quad \text{per le coordinate di P1}$$

$$\begin{cases} y = m2\,x + n2 \\ y^2 + x^2 = R^2 \end{cases}$$ per le coordinate di P2

Essendo D = 0 i due sistemi hanno le seguenti soluzioni:

Per P1 si ha $x1 = -\,m1\,n1 / (m1^2 + 1)$ $y1 = n1 / (m1^2 + 1)$

Per P2 si ha $x2 = -\,m2\,n2 / (m2^2 + 1)$ $y2 = n2 / (m2^2 + 1)$

La retta individuata da P1 e P2 ha equazione:

$$(y - y1) / (x - x1) = (y2 - y1) / (x2 - x1)$$

che sviluppata dà:

$$y = [(y2 - y1) / (x2 - x1)]\,x - x1\,[(y2 - y1) / (x2 - x1)] + y1$$

in cui $m = m3 = (y2 - y1) / (x2 - x1)$

$n = n3 = -\,x1\,[(y2 - y1) / (x2 - x1)] + y1$

A2.4 La compilazione del programma di calcolo

A questo punto non resta che trasformare in simbologia Qbasic ciò che abbiamo esplicitato in precedenza.
Per semplificare il lavoro di compilazione ci serviremo del programma dell'esercizio di paragrafo 3.28.1 relativo al tracciamento della circonferenza.
Aggiungeremo ad esso le istruzioni per l'inserimento delle coordinate di Po, del valore di R, dell'intervallo di variabilità della x, le istruzioni per il calcolo di (m1;n1), (m2;n2), (m3;n3), (x1;y1), (x2;y2), le istruzioni per il tracciamento delle due tangenti, le istruzioni per il tracciamento della retta passante per P1 e P2, nonché i comandi per la presentazione dei dati calcolati e per la presentazione grafica delle tre rette.

Le funzioni di calcolo sono computabili come segue:

per m1 e m2

```
m1 = (-(2*xo*yo) + SQR((2*xo*yo)^2 - 4*(R^2-xo^2)*(R^2-yo^2))/(2*(R^2-xo^2))

m2 = (-(2*xo*yo) - SQR((2*xo*yo)^2 - 4*(R^2-xo^2)*(R^2-yo^2))/(2*(R^2-xo^2))
```

per n1 e n2

```
n1 = yo-xo*m1

n2 = yo-xo*m2
```

per le coordinate dei punti di tangenza P1 e P2

x1 = - m1*n1/(m1^2+1)
y1 = n1/(m1^2+1)

x2 = - m2*n2/(m2^2+1)
y2 = n2/(m2^2+1)

le equazioni delle rette tangenti

yt1 = m1*x+n1

yt2 = m2*x+n2

l'equazione della retta passante per P1 e P2

m3 = (y2-y1)/(x2-x1)
n3 = (-x1*(y2-y1)/(x2-x1))+y1
y3 = m3*x+n3

l'equazione della circonferenza

C1 = SQR(R^2-x^2)
C2 = -SQR(R^2-x^2)

Con le corrispondenze simboliche sopra scritte compiliamo infine il programma di calcolo con la presentazione dei risultati numerici a sole due cifre decimali, più che sufficienti in questo tipo di esercizio:

```
SCREEN 9 ' impostazione del sistema grafico a 4 quadranti

FOR x = 0 TO 460 STEP 23

FOR y = 0 TO 320 STEP 2

PSET (x, y), 7

NEXT y

NEXT x

FOR y = 0 TO 320 STEP 16

FOR x = 0 TO 460 STEP 3

PSET (x, y), 7

NEXT x

NEXT y

LINE (0, 160)-(460, 160)

LINE (230, 0)-(230, 320)

FOR x = -10 TO 10 STEP .01                        &
```

```
NEXT x

LOCATE 1, 60: INPUT "xo="; xo ' richiesta introduzione ascissa di Po

LOCATE 2, 60: INPUT "yo="; yo ' richiesta introduzione ordinata di Po

LOCATE 3, 60: INPUT "R="; R ' richiesta introduzione raggio della circonferenza

LOCATE 4, 60: INPUT "Est="; e ' richiesta introduzione estensione intervallo assi cartesiani

LOCATE 5, 60: INPUT "step"; s ' richiesta introduzione incremento di calcolo

FOR x = -R TO R STEP s  '  impostazione del campo di variabilità della x per la circonferenza

C1 = SQR((R ^ 2) - (x ^ 2)) ' equazione della circonferenza (ramo superiore)

C2 = -SQR((R ^ 2) - (x ^ 2)) ' equazione della circonferenza (ramo inferiore)

PSET (230 + (230 / e) * x, 160 - (160 / e) * C1), 10 ' tracciamento ramo superiore di C in verde

PSET (230 + (230 / e) * x, 160 - (160 / e) * C2), 10 ' tracciamento ramo inferiore di C in verde

NEXT x ' rimanda all'istruzione FOR x = -R ...

' computazioni dei coefficienti angolari delle rette tangenti

m1 = (-(2 * xo * yo) + SQR((2 * xo * yo) ^ 2 - 4 * (R ^ 2 - xo ^ 2) * (R ^ 2 - yo ^ 2))) / (2 * (R ^ 2 - xo ^ 2))

m2 = (-(2 * xo * yo) - SQR((2 * xo * yo) ^ 2 - 4 * (R ^ 2 - xo ^ 2) * (R ^ 2 - yo ^ 2))) / (2 * (R ^ 2 - xo ^ 2))

n1 = yo - xo * m1 ' computazione dei parametri delle rette tangenti

n2 = yo - xo * m2

x1 = (-m1 * n1) / (m1 ^ 2 + 1) ' computazione coordinate punti di tangenza

x2 = (-m2 * n2) / (m2 ^ 2 + 1)

y1 = n1 / (m1 ^ 2 + 1)

y2 = n2 / (m2 ^ 2 + 1)

m3 = (y2 - y1) / (x2 - x1)  ' computazione coefficiente angolare retta passante per P1 e P2

n3 = -(x1 * (y2 - y1) / (x2 - x1)) + y1 ' computazione parametro retta passante per P1 e P2

FOR x = -e TO e STEP s ' impostazione del campo di variabilità della x per le tre rette

yt1 = m1 * x + n1 ' equazione prima retta tangente

yt2 = m2 * x + n2 ' equazione seconda retta tangente

y3 = m3 * x + n3 ' equazione retta passante per P1 e P2

PSET (230 + (230 / e) * x, 160 - (160 / e) * yt1), 14 ' tracciamento prima retta tangente in giallo

PSET (230 + (230 / e) * x, 160 - (160 / e) * yt2), 14 ' tracciamento seconda retta tangente in giallo

PSET (230 + (230 / e) * x, 160 - (160 / e) * y3), 3 ' tracciamento  retta passante per P1 e P2 in turchese      &
```

```
NEXT x ' rimanda all'istruzione For x= -e....

LOCATE 10, 60: PRINT "m1="; USING "###.##"; m1 ' presentazione dati calcolati
                                                ' a 3 interi e 2 decimali
LOCATE 11, 60: PRINT "n1="; USING "###.##"; n1

LOCATE 12, 60: PRINT "m2="; USING "###.##"; m2

LOCATE 13, 60: PRINT "n2="; USING "###.##"; n2

LOCATE 14, 60: PRINT "x1="; USING "###.##"; x1

LOCATE 15, 60: PRINT "y1="; USING "###.##"; y1

LOCATE 16, 60: PRINT "x2="; USING "###.##"; x2

LOCATE 17, 60: PRINT "y2="; USING "###.##"; y2

LOCATE 18, 60: PRINT "m3="; USING "###.##"; m3

LOCATE 19, 60: PRINT "n3="; USING "###.##"; n3
```

Utilizziamo il programma per la ricerca delle due rette passanti per Po, di coordinate xo = 4; yo = 7, tangenti alla circonferenza C di raggio R = 5.
Essendo verificata la disuguaglianza |yo| > R il problema è possibile.
Imponiamo al grafico una estensione (Est) degli assi cartesiani tale da rendere la visualizzazione completa di C e Po:
deve essere (Est) > |xo|; |yo|; R, si assume, per calibrare anche le scale, (Est) = 10 con s = .02 pari a 1000 punti di calcolo.
Premendo F5 si ha la richiesta di introduzione dati; dopo l'ultimo valore inserito si ottiene la presentazione di tutti gli elementi calcolati e la soluzione grafica del nostro problema così come mostrato in figura 78.

La schermata del P.C., riportata in figura, contiene quanto richiesto nella proposizione del problema:
-i valori di m1; n1 e m2; n2 per la costruzione analitica delle equazioni delle due rette tangenti la circonferenza
-le coordinate dei due punti di tangenza tra le rette e la circonferenza (x1 ; y1), (x2 ; y2)
-i valori di m3; n3 per la costruzione analitica della equazione della retta passante per P1 e P2
-il grafico risolutivo del problema tracciato in un sistema di assi cartesiani aventi scale calibrate, con intervalli in x e y ciascuno suddiviso in 10 unità

Le equazioni delle rette sono pertanto:

le tangenti

$$y = .4\,x + 5.39$$

$$y = -\,6.62\,x + 33.5$$

la retta passante per P1 e P2

$$y = .57\,x + 3.57$$

Per completare l'esercizio si deve osservare che se i punti P1 e P2 avessero ascisse uguali, x1 = x2, le espressioni per il calcolo di m3 ed n3 non sarebbero algebricamente computabili perché avrebbero i denominatori nulli.

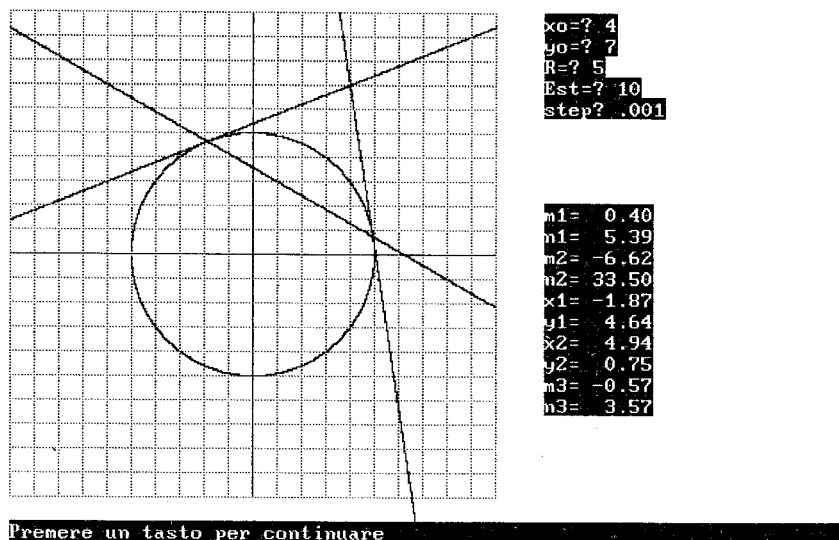



Figura 78
Schermata risolutiva del problema dato

A2.5 Note

L'esercizio che abbiamo svolto, essendo a parametri variabili, propone l'impiego del programma in tutti i casi simili avendo come gradi di libertà le coordinate (xo ; yo) di Po e il raggio R della circonferenza.
Si consiglia di ripetere la computazione per diversi valori dei parametri per poter comprendere quale è l'efficacia del programma sviluppato.
Naturalmente per la soluzione di un caso più generale del problema, relativo al calcolo delle tangenti ad una circonferenza ovunque collocabile nel piano, è necessaria una più complicata impostazione analitica, seguendo però le tracce del lavoro già svolto; lavoro che può servire inoltre come guida per la soluzione di problemi di geometria analitica riguardanti curve diverse da quelle ora trattate.

BIBLIOGRAFIA

-HANDBOOK OF MATHEMATICAL FUNCTIONS
Abramowitz and Stegun Dover Publications, Inc. New York

-MANUALE MS-DOS Q BASIC
Microsoft Corporation

-PROGRAMMARE IN QUICKBASIC Don Inmann, Bob Albrecht
Mc Graw-HiII Italia Milano

-METODI MATEMATICI NELL'INGEGNERIA T.v.Karman, MA. Biot
Einaudi Torino

-ESERCIZI DI ANALISI MATEMATICA vol.I B.Finzi F.Morra
Libreria Editrice Politecnica Cesare Tamburini Milano

-ANALISI MATEMATICA volI. G.Moretti
Hoepli Milano

-ANALISI MATEMATICA volII parte prima G.Moretti
Hoepli Milano

-ANALISI MATEMATICA volII parte seconda G .Moretti
Hoepli Milano

-THE FOURIER INTEGRAL AND ITS APPLICATIONS A.Papoulis
Mc Graw-HiII New York

-What is the Fast Fourier Transform? W.T.Cochran, J.W. Coo]ey
IEEE Trans. Audio Electroacoust. vol. AU-15,pp.45-55, June 1967

-LA CORRELAZIONE C.DeI Turco La Moderna La Spezia

-IL PROBLEMA GEOMETRICO E LA GEOMETRIA ANALITICA R.Ferrauto
Soc. Editrice Dante Alighieri Città di Castello